

PDF VERSION BY



ParsTech

2007

www.parstech.org

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

برنامه نویسی

(الگوریتم، فلوچارت و زبان VB)

قسمت اول

شاخه کاردانش

استاندارد مهارت: رایانه کار درجه ۱

شماره استاندارد: ۴۲/۲۷-۳

شماره درس: ۸۹۹۷-۸۹۹۶

الگوریتم و فلوچارت [کتاب های درسی] شماره استاندارد ۴۲/۲۷-۳ / برنامه ریزی محتوا و نظارت بر تألیف دفتر برنامه ریزی و تألیف آموزش های فنی و حرفه ای و کاردانش، مؤلف منصور ولی نژاد؛ [برای] وزارت آموزش و پرورش سازمان پژوهش و برنامه ریزی آموزشی، - تهران: دیپاگران تهران، ۱۳۸۴.
۹۲ ص. - (شاخه کاردانش استاندارد مهارت رایانه کار درجه ۱؛ شماره درس ۸۹۹۷-۸۹۹۶)
ISBN: 964-354-581-4

فهرست نویسی براساس اطلاعات فیپا.

۱. الگوریتمها. ۲. نمودار جریان کار. الف. ولی نژاد، منصور، ۱۳۴۵ - ب. سازمان پژوهش و برنامه ریزی آموزشی، دفتر تألیف و برنامه ریزی درسی آموزش های فنی و حرفه ای و کاردانش، سازمان پژوهش و برنامه ریزی آموزشی، ج. سازمان پژوهش و برنامه ریزی آموزشی.

۴۲/۲۷-۳ ۳۷۳

الف ۹ / ۵۸ QA

۱۳۸۴

۸۴-۱۱۶۷۰م

کتابخانه ملی ایران

همکاران محترم و دانش‌آموزان عزیز:

پیشنهادها و نظرات خود را درباره محتوای این کتاب به نشانی تهران - صندوق پستی شماره ۴۸۷۴/۱۵ دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش یا سایت: WWW.TVOCCD.SCH.IR ارسال فرمایید.

وزارت آموزش و پرورش

سازمان پژوهش و برنامه‌ریزی آموزشی

برنامه‌ریزی محتوا و نظارت بر تألیف: دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش

نام کتاب مهارتی: برنامه‌نویسی (الگوریتم، فلوچارت و زبان VB) قسمت اول

مؤلف: مهندس منصور ولی‌نژاد

کمیسیون تخصصی: مهندس علی رضا جباریه - مهندس ملیحه طرزی -

مهندس شهناز علی‌زاده - دکتر بتول عطاران - مهندس حمید احمدی

ویراستار: مجتمع فنی تهران (هما تیموری)

اجرای کامپیوتری: مجتمع فنی تهران (معصومه باقری)

طراح جلد: مجتمع فنی تهران (محبوبه توکلی)

نوبت چاپ و سال انتشار: اول - تیرماه ۱۳۸۴

چاپخانه: اتاق چاپ

تیراژ: ۸۵۰۰۰

نظارت بر چاپ: اداره کل چاپ و توزیع کتابهای درسی

ناشر: مؤسسه فرهنگی هنری دیباگران تهران با همکاری وزارت آموزش و پرورش

تهران - سعادت آباد - میدان کاج - خیابان سرو شرقی - روبه‌روی خیابان علامه - پلاک ۹۷

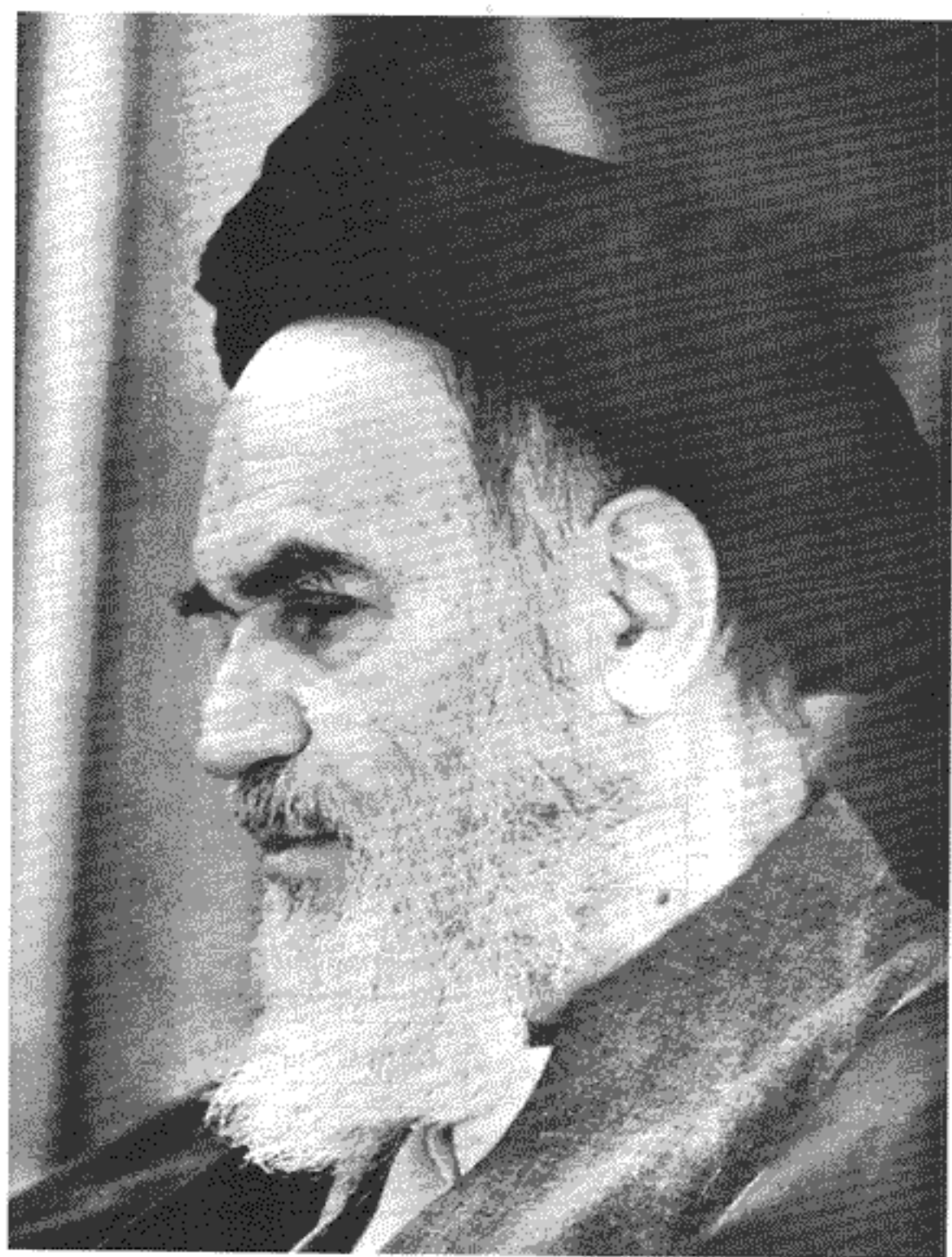
تلفن: ۲۲۰۹۸۴۴۶-۷ دورنگار: ۲۲۰۹۸۴۴۸ صندوق پستی: ۱۴۶۵۵/۴۶۶

این کتاب براساس استاندارد مهارتی رایانه کاردرجه ۱ به سفارش دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش سازمان پژوهش و برنامه‌ریزی آموزشی وزارت آموزش و پرورش توسط مؤسسه فرهنگی هنری دیباگران تهران تألیف و پس از تصویب دفتر مذکور به چاپ رسیده است.

*** حق چاپ محفوظ است ***

ISBN: 964-354-581-4

شابک: ۹۶۴-۳۵۴-۵۸۱-۴



بدانید مادام که در احتیاجات صنایع پیشرفته، دست خود را پیش دیگران دراز کنید و به درיوزگی عمر را بگذرانید، قدرت ابتکار و پیشرفت در اختراعات در شما شکوفا نخواهد شد.

امام خمینی «قدس السره الشریف»

استاندارد آموزشی

رایانه کار درجه ۱

سیستم عامل پیشرفته

برنامه نویسی
(الگوریتم، فلوچارت و زبان VB)

نرم افزار NU

مفاهیم شبکه

رایانه کار درجه ۲

مبانی و فناوری کامپیوتر

سیستم عامل مقدماتی
(DOS-Windows XP)

واژه پرداز Word 2002

صفحه گسترده Excel 2002

ارایه مطالب PowerPoint 2002

بانک اطلاعاتی Access 2002

اینترنت، پست الکترونیکی و
ویروسهای کامپیوتری

فهرست مطالب

۷.....	مقدمه ناشر.....
۸.....	مقدمه.....

واحدکار ۱: توانایی حل مسایل و طراحی الگوریتم مناسب برای آن‌ها

۱۰.....	کلیات.....
۱۰.....	۱-۱ شناخت مسایل و آرایه راه حل مناسب برای حل آن‌ها.....
۱۱.....	۱-۲ الگوریتم.....
۱۱.....	۱-۲-۱ تعریف الگوریتم.....
۱۲.....	۱-۲-۲ شرایط الگوریتم.....
۱۳.....	۱-۲-۳ انواع دستورالعمل‌ها در الگوریتم.....
۳۷.....	۱-۳ آرایه.....
۴۳.....	۱-۴ زیرالگوریتم.....
۴۶.....	۱-۵ روش‌های مرتب‌سازی داده‌ها و اطلاعات.....
۴۶.....	۱-۵-۱ روش مرتب‌سازی حبابی (Bubble Sort).....
۴۸.....	۱-۶ روش‌های جستجوی داده‌ها و اطلاعات.....
۴۸.....	۱-۶-۱ روش جستجوی خطی.....
۴۹.....	۱-۶-۲ روش جستجوی باینری (دودویی).....
۵۱.....	۱-۷ پشته (Stack).....
۵۷.....	۱-۸ صف (Queue).....
۵۹.....	خلاصه مطالب.....
۶۰.....	آزمون نظری.....

واحدکار ۲: توانایی ترسیم فلوچارت

۶۴.....	کلیات.....
۶۴.....	۲-۱ علایم و اشکال در فلوچارت.....
۶۴.....	۲-۱-۱ علایم شروع و پایان.....
۶۴.....	۲-۱-۲ علایم ورودی.....
۶۵.....	۲-۱-۳ علامت انتساب.....

۶۵	 ۲-۱-۴ علامت شرط
۶۶	 ۲-۱-۵ علایم خروجی
۶۷	 ۲-۱-۶ علامت اتصال
۸۶	 خلاصه مطالب
۸۷	 آزمون نظری
۹۱	 آزمون پایانی
۹۲	 پاسخنامه
۹۲	 فهرست منابع

سپاس بیکران پروردگار را که به انسان قدرت اندیشیدن بخشید تا به یاری این موهبت راه ترقی و تعالی را ببیماید و امید به این که عنایات الهی شامل حال ما باشد تا با بضاعت اندک علمی خود در خدمت جوانان و آینده سازان کشور عزیزمان باشیم.

یکی از بارزترین ویژگی‌های عصر حاضر، حضور گسترده کامپیوتر در کلیه عرصه‌های فعالیت انسان است به گونه‌ای که انجام برخی از کارها، بدون استفاده از کامپیوتر قابل تصور نیست. کامپیوتر به عنوان ابزاری قدرتمند، سرعت و دقت کارها را فوق العاده افزایش داده و گذرگاه‌های صعب‌العبور علم را به شاهراه‌های هموار مبدل ساخته است. به همین دلیل در جهان کنونی، آموزش و یادگیری علوم کامپیوتر یک ضرورت اجتناب‌ناپذیر است.

در همین راستا دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش براساس موافقت‌نامه‌ای، تألیف کتاب‌های مهارت‌های رایانه کار درجه ۱ و ۲ شاخه کاردانش را به مؤسسه فرهنگی هنری دیباگران تهران محول کرده که افتخاری بزرگ است. کتاب حاضر با همکاری جمعی از اساتید، متخصصان و مهندسان مجرب رشته کامپیوتر تألیف و محتوای آن در کمیسیون تخصصی برنامه‌ریزی و تألیف کتاب‌های درسی رشته کامپیوتر مورد تأیید و توسط دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش مورد بررسی و تصویب قرار گرفته است.

طراحی کتاب‌ها براساس ساختار آموزش‌های پیمانه‌ای (مادولار) انجام گرفته و ساختار آن براساس توانایی‌های مورد انتظار در استانداردهای مهارتی طراحی شده است. این کتاب‌ها حتی‌المقدور به صورت خودآموز و خودمحتوا سازمان‌دهی شده است و تلاش بر این است که کتاب‌های آموزش گام به گام، به همراه مثال‌ها، تمرین‌های عملی و کاربردی برای کارهای آزمایشگاهی و کارگاهی به انضمام سؤالات پیش‌آزمون و آزمون پایانی، مجموعه منسجمی از هر پیمانه را آرایه دهد به طوری که دانش‌آموزان پس از پایان هر پیمانه، از مهارت کافی برای کار با موضوع پیمانه برخوردار باشند.

در خاتمه از حسن نظر و اعتماد دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش سازمان پژوهش و برنامه‌ریزی آموزشی و سایر همکاران در مؤسسه فرهنگی هنری دیباگران تهران سپاسگزاری می‌کنیم و امیدواریم نقشی هر چند کوچک در جهت اشاعه فناوری اطلاعاتی که محور توسعه در جهان است، ایفا کرده باشیم. ضمناً یادآوری می‌شود که استانداردهای مهارت رایانه کار درجه ۱ و ۲ توسط سازمان آموزش فنی و حرفه‌ای وزارت کار و امور اجتماعی با عنایت به تغییرات حوزه معرفتی علم رایانه، بازنگری و تجدیدنظر شده است و این تألیف حاصل تجدیدنظر فوق‌الذکر می‌باشد.

در جوامع بشری یکی از مسایلی که همواره ذهن انسان را متوجه خود کرده، یافتن راه‌حلی مناسب برای مسائل مختلف و انجام کارهای موردنظر خود بوده و به این منظور بیشتر از تجربیات قبلی و حدسیات خود استفاده کرده است. با ظهور و پیشرفت علوم چون ریاضیات و نجوم، نیاز به روش‌های جدیدتر و استفاده از راه‌حل‌های ریاضی و منطقی بیشتر مورد توجه دانشمندان قرار گرفت. برای اولین بار خوارزمی به این نکته اشاره کرد و مبنای روش حل مسائل با استفاده از روش‌های ریاضی و منطقی را بنیان گذاشت. بعدها اروپاییان مبنای ارایه شده توسط خوارزمی را در حل مسائل علمی خود مورد استفاده قرار دادند و نام الگوریتم را (با توجه به عبارت خوارزمی) برای آن برگزیدند. با اختراع اولین کامپیوترها و نیاز به برنامه‌نویسی برای حل مسایلی که به کامپیوتر محول می‌شد، این روش بیشتر مورد توجه قرار گرفت و برنامه‌نویسان سال‌های متمادی از آن استفاده کردند.

چون الگوریتم براساس ارایه یک راه‌حل به وسیله زبان محاوره‌ای انسان بنا شده است، در مسائل پیچیده که الگوریتم‌های طولانی را در پی دارد برنامه‌نویسان را دچار سردرگمی و اشتباه می‌کند. بنابراین روش دیگری به نام فلوچارت در کنار الگوریتم مورد استفاده قرار می‌گیرد تا با استفاده از خطوط و اشکال مختلف هندسی درک مسیر اجرا و نوع دستورالعمل‌هایی که در الگوریتم به کار می‌رود، ساده‌تر شود.

امروزه با وجود روش‌های نوین برنامه‌نویسی و زبان‌های برنامه‌نویسی توانمند، استفاده از الگوریتم و فلوچارت در تهیه نرم‌افزارها به میزان قابل توجهی کاسته شده است. با این حال به کارگیری این مباحث تا حد زیادی می‌تواند در پرورش تفکر ریاضی و منطقی در کسی که روش‌های برنامه‌نویسی را می‌آموزد، مؤثر واقع شود.

در این کتاب سعی بر این است که دانش‌آموزان با مفاهیم اساسی در الگوریتم و فلوچارت آشنا شده و چگونگی یافتن راه‌حل مناسب برای حل مسائل مختلف را به همراه مثال و تمرین‌های متعدد فراگیرند. به علاوه مفاهیم پیشرفته‌تری مانند روش‌های مرتب‌سازی، جستجو و غیره برای رشته برنامه‌نویسی به صورت مجزا ارایه شده است.

در پایان از تمام اساتید، معلمان و دانش‌پژوهان تقاضا دارم تا اینجانب را از نظرات، پیشنهادات و انتقادات خود بهره‌مند سازند.

مؤلف

هدف کلی

توانایی نوشتن الگوریتم و ترسیم فلوچارت مسائل

هدف جزئی



توانایی حل مسائل و طراحی الگوریتم مناسب
برای آن‌ها

زمان (ساعت)	
عملی	نظری
۱۰	۲۰

هدفهای رفتاری ▼

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱- نحوه شناخت و بررسی مسائل مختلف را توضیح دهد.
- ۲- مفهوم الگوریتم را تعریف کند و ویژگی‌های آن را بیان نماید.
- ۳- انواع دستورالعمل‌ها را در الگوریتم بیان کند و کاربرد آن‌ها را توضیح دهد.
- ۴- مفهوم عملگر را بداند و انواع عملگرها را بیان کند.
- ۵- عملگرهای ریاضی، منطقی و مقایسه‌ای و کاربرد آن‌ها را بیان کند و حق تقدم آن‌ها را نسبت به یکدیگر توضیح دهد.
- ۶- بتواند انواع الگوریتم‌ها را برای مسائل متفاوت طراحی نماید.
- ۷- مفهوم آرایه‌های یک بعدی و دو بعدی را توضیح دهد و بتواند از آن‌ها استفاده کند.
- ۸- مفهوم زیرالگوریتم و روش‌های مرتب‌سازی و جستجوی داده‌ها را توضیح دهد و بتواند از آن‌ها استفاده کند.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

کلیات

روش حل مسایل با استفاده از روش‌ها و تحلیل‌های ریاضی و منطقی اولین بار به وسیله دانشمند بزرگ ایرانی - خوارزمی - مورد توجه قرار گرفت، وی علاوه بر مباحث مختلفی که در علوم ریاضی و نجوم طرح کرد پایه و اساس روشی را در حل مسایل بنیان گذاشت که در آینده مدت‌ها مورد استفاده برنامه‌نویسان کامپیوتر قرار گرفت. نام الگوریتم نیز به افتخار وی و از عبارت "الخوارزمی" گرفته شده است.

در این پیمانه مهارتی با نحوه تجزیه و تحلیل انواع مختلف مسایل و جستجو و طراحی راه حل‌های مناسب برای آن‌ها آشنا خواهید شد و در پایان آرایه راه حل به دست آمده با توجه به قواعد علم الگوریتم را فرا خواهید گرفت.

۱-۱ شناخت مسایل و آرایه راه حل مناسب برای حل آن‌ها

به منظور آرایه راه حل مناسب برای یک مسأله، بهترین کار بررسی آن یا به عبارت دیگر تجزیه و تحلیل مسأله است. به این منظور و برای شناخت بهتر یک مسأله باید سه عامل مهم را در نظر بگیرید؛ این عوامل عبارتند از: مقادیر معلوم، خواسته‌های مسأله (مجهولات) و عملیات محاسباتی. در این جا این پارامترها را مورد بررسی قرار می‌دهیم.

مقادیر معلوم: مقادیری که در اختیار مسأله قرار می‌گیرند و برای رسیدن به هدف موردنظر در مسأله موردنیاز هستند.

محاسبات: برای رسیدن به نتایج موردنظر معمولاً لازم است تا عملیاتی را روی مقادیر معلوم انجام دهید؛ بخش عمده‌ای از این عملیات معمولاً با استفاده از فرمول‌های مختلف انجام می‌شود؛ البته محاسبات می‌توانند با توجه به روابط منطقی که بین مقادیر معلوم و خواسته‌های مسأله وجود دارند، انجام گیرند.

خواسته‌های مسأله (مجهولات): مقادیری هستند که معمولاً در اثر انجام عملیات روی مقادیر معلوم حاصل می‌شوند، البته مجهولات می‌توانند از روابط منطقی که در حل مسأله دخالت می‌کنند نیز به وجود آمده و مورد استفاده قرار گیرند.

به عنوان مثال فرض کنید می‌خواهیم محیط و مساحت یک دایره به شعاع دلخواه را محاسبه کنیم. برای حل این مسأله و آرایه راه حل مناسب با روش آرایه شده، ابتدا مقادیر معلوم برای حل این مسأله را مورد توجه قرار می‌دهیم؛ همان‌طور که می‌دانید برای محاسبه محیط و مساحت هر دایره باید شعاع آن را در اختیار داشته باشیم، بنابراین شعاع دایره (R) به عنوان تنها داده موردنیاز برای حل مسأله کافی است.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

محاسباتی که برای رسیدن به محیط و مساحت دایره لازم است، درواقع فرمول‌های زیر خواهند بود:

$$P = 2 \times 3/14 \times R$$

$$S = 3/14 \times R^2$$

همان‌طور که می‌بینید با استفاده از این فرمول‌ها روابط بین داده‌های ورودی و نتایج موردنظر به خوبی تعیین می‌شود و بالاخره به عنوان مرحله آخر، همان‌گونه که از صورت مسأله و روابط ریاضی آرایه شده برداشت می‌شود خواسته‌های مسأله (مجهولات) همان محیط (P) و مساحت (S) دایره هستند که در متغیرهای مربوطه ذخیره می‌شوند.

تمرین: مقادیر معلوم، محاسبات و خواسته‌های مسأله (مجهولات) را در حل یک مسأله معادله درجه دو معین کنید.

۲-۱ الگوریتم

پس از بررسی مسأله و شناخت سه عامل اصلی برای حل آن لازم است تا روشی برای حل مسأله آرایه شود. این روش‌ها می‌توانند با استفاده از تجارب قبلی در حل مسایل و به دلخواه کسی که مسأله را حل می‌کند، به دست آید یا با استفاده از روش‌های علمی مبتنی بر تحلیل‌های ریاضی و منطقی صورت پذیرد. الگوریتم در واقع مفهومی است که مسایل را با استفاده از تحلیل‌های ریاضی و منطقی مورد بررسی قرار داده و راه حل مناسبی برای آن آرایه می‌کند.

۱-۳-۱ تعریف الگوریتم

به مجموعه‌ای از دستورالعمل‌ها که با زبان دقیق و قابل فهم به همراه جزییات لازم و به صورت مرحله به مرحله به گونه‌ای اجرا شده که هدف خاصی (حل مسأله) را دنبال کنند و شروع و خاتمه آن‌ها نیز مشخص باشد، الگوریتم می‌گویند. به عبارت دیگر می‌توان الگوریتم را به یک ماشین تشبیه کرد که مقادیر معلوم را دریافت کرده، روی آن‌ها محاسباتی را انجام می‌دهد و در پایان خواسته‌های مسأله (مجهولات) را آرایه می‌کند.

همان‌طور که می‌بینید رابطه نزدیکی بین مفهوم الگوریتم و نحوه کار کامپیوتر در حل مسایل وجود دارد، بنابراین با استفاده از روش الگوریتم می‌توانید حل مسایل را به گونه‌ای طراحی کنید که برای تبدیل به زبان کامپیوتر نیز قابل فهم باشد. به‌طور معمول با مفهوم الگوریتم آشنا هستید و از آن استفاده می‌کنید، به عنوان مثال وقتی هر روز برای کسب علم و دانش به مدرسه می‌روید، اعمالی را به ترتیب و به صورت دقیق و کامل انجام می‌دهید یعنی ابتدا از خواب بیدار می‌شوید، دست و صورت خود را می‌شوید، صبحانه می‌خورید و سپس لباس مناسب به تن کرده و بعد از برداشتن وسایل و کتاب‌های

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

موردنیاز از در خانه خارج شده مسافتی را طی کرده و به مدرسه و کلاس خود می‌روید. اگر هر یک از این اعمال را قبل یا بعد از عمل دیگر انجام دهید هدف موردنظر که شکل درست به مدرسه رفتن است، انجام نمی‌شود. اکنون می‌توان این مراحل را به صورت زیر و به شکل خلاصه‌تر و قابل فهم‌تری بیان کرد:

- بیدار شدن از خواب
- شستن دست و صورت
- خوردن صبحانه
- پوشیدن لباس مدرسه
- برداشتن کتاب‌های درسی و دفتر و قلم
- خروج از خانه
- رفتن به مدرسه و ورود به کلاس درس

تمرین: مراحل تهیه نان در نانوائی را به صورت مرحله به مرحله بنویسید.

۲-۳-۱ شرایط الگوریتم

اکنون که با مفهوم الگوریتم آشنا شدید، لازم است ویژگی‌های یک الگوریتم را بشناسید.

الف - استفاده از زبان ساده، دقیق و قابل فهم: این ویژگی سبب می‌شود تا در انجام دستورالعمل‌ها همواره یک برداشت یکسان حاصل شود، در غیراین صورت برداشت‌های متفاوت سبب خواهد شد تا دستورالعمل‌ها نتایج متفاوتی را به همراه داشته باشند. زبان الگوریتم نیز می‌تواند یکی از زبان‌های گفتاری و نوشتاری مانند فارسی، انگلیسی و نظایر آن باشد.

ب - استفاده از جزییات کافی: این ویژگی سبب می‌شود تا دستورالعمل‌ها به‌طور کامل اجرا شوند. وجود موارد نامشخص یا ارایه دستورالعمل‌ها به صورت کلی و مبهم سبب مخدوش شدن نتایج خواهد شد.

ج - شروع و پایان الگوریتم: در یک الگوریتم باید شروع دستورالعمل‌ها مشخص باشد. هر الگوریتم یک نقطه شروع دارد که به عنوان اولین دستورالعمل از آن استفاده می‌شود، به علاوه پایان و خاتمه الگوریتم نیز باید تعیین شود. یک الگوریتم می‌تواند بیش از یک نقطه پایان داشته باشد.

د - ترتیب انجام دستورالعمل‌ها: یکی از ویژگی‌های مهم یک الگوریتم ترتیب اجرای دستورالعمل‌ها است؛ اگر این کار به درستی انجام نشود، پیش‌بینی نتیجه کار مشخص نخواهد بود. در یک الگوریتم ترتیب انجام عملیات با استفاده از شماره‌گذاری دستورالعمل‌ها از بالا به پایین انجام می‌شود که البته در

استاندارد مهارت رایانه کار درجه ۱	پیمانه مهارتی الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

صورت نیاز می‌توان ترتیب اجرای دستورالعمل‌ها را نیز تغییر داد. در مباحث بعد به این مسأله خواهیم پرداخت.

در این‌جا ذکر این نکته لازم است که اگر در حل مسایل سه عامل اصلی را به دقت مشخص کنید، طراحی یک الگوریتم کار زیاد دشواری نخواهد بود. اگر مقادیر معلوم، خواسته‌های مسأله، فرمول‌ها و روابط ریاضی و منطقی بین آن‌ها را به درستی تعیین کنید، نوشتن الگوریتم‌های مختلف آسان‌تر خواهد شد.

هـ - جامع بودن: الگوریتم باید به شکلی طراحی شود تا با توجه به صورت مسأله و مفروضات آن در تمام حالت‌ها، نتایج مناسب و صحیحی را ارائه کرده و در حالت‌های خاص یا داده‌های ورودی متفاوت، نتایج درستی را ایجاد کند.

و - استفاده از حداقل دستورالعمل‌ها: در یک الگوریتم باید از دستورات اضافه که سبب افزایش حجم الگوریتم می‌شود، خودداری نمایید چرا که این کار الگوریتم را شلوغ کرده و باعث سردرگمی می‌شود.

۳-۲-۱ انواع دستورالعمل‌ها در الگوریتم :

همان‌طور که می‌دانید الگوریتم مجموعه‌ای از دستورالعمل‌هاست. دستورالعمل‌ها انواع مختلفی دارند که در این‌جا به ذکر تعدادی از آن‌ها می‌پردازیم:

- الف - دستورالعمل‌های ورودی
- ب - دستورالعمل‌های خروجی
- ج - دستورالعمل‌های محاسباتی
- د - دستورالعمل‌های شرطی
- هـ - دستورالعمل‌های تکرار (حلقه‌ها)

الف - دستورالعمل‌های ورودی

این دستورالعمل‌ها برای دریافت داده‌های ورودی استفاده می‌شوند و معمولاً برای اجرای آن‌ها از عباراتی مانند "بخوان، دریافت کن یا بگیر" استفاده می‌شود.

ب - دستورالعمل‌های خروجی

این دستورالعمل‌ها برای نمایش اطلاعات خروجی یا پیام‌های موردنیاز به منظور راهنمایی کاربر روی صفحه نمایش یا چاپ آن‌ها به وسیله چاپگر استفاده می‌شوند و معمولاً برای اجرای آن‌ها عباراتی

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۳۱

مانند "نمایش بده یا چاپ کن" به کار می‌روند.

ج - دستورالعمل‌های محاسباتی

این دستورالعمل‌ها در واقع نحوه آرایه و استفاده از فرمول‌ها و انجام عملیات ریاضی و محاسباتی را تعیین می‌کنند و معمولاً برای اجرای آن‌ها از همان شکلی که در ریاضیات وجود دارد، استفاده می‌شود یعنی در سمت راست تساوی عملیات محاسباتی و در سمت چپ تساوی نام یک متغیر به کار می‌رود؛ البته به جای علامت تساوی از علامت فلش ($\leftarrow$) نیز استفاده می‌شود.

تعریف متغیر: متغیرها مکان‌هایی هستند که توانایی نگهداری و ذخیره‌سازی انواع داده را دارند. متغیرها در الگوریتم همان کاربرد ریاضی خود را دارا هستند و علاوه بر اعداد می‌توانند متن یا مقادیر منطقی درست یا نادرست، تاریخ، ساعت و نظایر آن‌ها را نگهداری کنند.

نحوه استفاده از عملگرهای ریاضی در الگوریتم

همان‌طور که اشاره شد برخی از عملیاتی که در الگوریتم انجام می‌گیرد، عملیات ریاضی و محاسباتی است که برای انجام این‌گونه دستورات لازم است از عملگرهای ریاضی استفاده شود.

تعریف عملگر: در واقع عملگر یک یا مجموعه عملیاتی است که در ماشین الگوریتم به صورت از پیش آماده تعریف شده است و برای استفاده از آن کافی است از علایمی که به صورت قراردادی تعریف شده است، استفاده کنید مانند عملگر جمع که می‌تواند دو عدد یا دو متغیر یا ترکیبی از آن‌ها را با هم جمع و حاصل آن را آرایه کند. اعداد و متغیرهایی که عملگر روی آن‌ها عملیات انجام می‌دهد، عملوند نامیده می‌شوند. عملگرهای ریاضی در جدول ۱-۱ آرایه شده‌اند.

جدول ۱-۱

عملگر	عنوان	مثال
()	پرانتز	$(2 \times 3) + 4 = 10$
$\times$	ضرب	$2 \times 3 = 6$
/	تقسیم	$12 / 6 = 2$
+	جمع	$35 + 14 = 49$
-	تفریق	$10 - 7 = 3$

نکته حایز اهمیت این است که عملگرهای آرایه شده در جدول ۱-۱ نسبت به هم دارای حق تقدم در اجرا هستند، به عبارت دیگر در یک عبارت ریاضی که از چند عملگر استفاده شده است عملگرها از سمت چپ عبارت ریاضی به ترتیب اجرا می‌شوند. اما همیشه این‌گونه نیست و در هنگام

استاندارد مهارت: رابطه کار درجه ۱	بیمه مهارت: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

استفاده از چند عملگر ریاضی، اولویت آن‌ها مطابق جدول ۱-۱ از بالا به پایین در نظر گرفته می‌شود، به عنوان مثال عبارت $2 + 3 \times 5 / 10$ را در نظر بگیرید؛ در این عبارت براساس جدول ۱-۱ ابتدا عدد ۳ در ۵ ضرب شده و سپس حاصل ضرب آن یعنی ۱۵، بر عدد ۱۰ تقسیم می‌شود چرا که عملگرهای ضرب و تقسیم نسبت به عملگر جمع حق تقدم دارند. در پایان، حاصل تقسیم یعنی عدد $10/5$ با عدد ۲ جمع و در نتیجه عدد $3/5$ به دست می‌آید.

در این جا ذکر این نکته نیز لازم است که عملگرهای ضرب، تقسیم، جمع و تفریق نسبت به هم دارای اولویت یکسانی هستند و به عبارت دیگر اولویت آن‌ها نسبت به هم از چپ به راست است. همین وضعیت در رابطه با عملگرهای جمع و تفریق نیز وجود دارد، به عنوان مثال عبارت $10 + 40 - 6 \times 2$ را در نظر بگیرید؛ در این عبارت ابتدا عدد ۶ در ۲ ضرب شده، سپس عبارت $10 + 40$ محاسبه می‌شود که نتیجه آن عدد ۵۰ خواهد بود و بعد عمل تفریق انجام می‌شود و حاصل تفریق $50 - 12$ یعنی ۳۸ به دست می‌آید.

مثال ۱: فرض کنید می‌خواهید مسأله محاسبه محیط و مساحت یک دایره به شعاع دلخواه R را به صورت یک الگوریتم بنویسید، قبلاً این مسأله را مورد بررسی قرار داده‌اید و سه عامل مقادیر معلوم، فرمول‌های محاسباتی و خواسته‌های مسأله (مجهولات) و رابطه بین آن‌ها را تعیین کرده‌اید، بنابراین الگوریتم حل این مسأله به صورت زیر خواهد بود:

توضیح	
شروع الگوریتم	۱- شروع
ورود داده‌ها	۲- R را دریافت کن
محاسبه محیط	۳- $P \leftarrow 2 \times 3.14 \times R$
محاسبه مساحت	۴- $S \leftarrow 3.14 \times R \times R$
نمایش خروجی	۵- P و S را نمایش بده
پایان الگوریتم	۶- پایان

تمرین: ترتیب انجام عملیات و حاصل عبارات زیر را تعیین کنید:

$$(14 - 3 \times 2) / 5 + 2$$

$$((27 + 4) \times 12 - 7) + 8 / 2$$

تمرین: الگوریتمی بنویسید که حجم یک منبع آب مکعب شکل را محاسبه کرده و نمایش دهد.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

مثال ۲: الگوریتمی بنویسید تا میانگین سه عدد دلخواه را محاسبه کرده و نمایش دهد.

ابتدا مسأله را مورد بررسی قرار دهید؛ در این مثال ورودی‌ها شامل سه عدد هستند و محاسبات لازم جهت به دست آوردن میانگین آن‌ها عبارتند از:

$$\text{میانگین} = \frac{\text{مجموع سه عدد}}{۳}$$

بنابراین الگوریتم مربوطه بدین صورت خواهد بود:

توضیح	
۱- شروع	شروع الگوریتم
۲- X را دریافت کن	ورود داده
۳- Y را دریافت کن	ورود داده
۴- Z را دریافت کن	ورود داده
۵- $SUM \leftarrow X + Y + Z$	محاسبه مجموع
۶- $AVE \leftarrow \frac{SUM}{۳}$	محاسبه میانگین
۷- AVE را نمایش بده	نمایش خروجی
۸- پایان	پایان الگوریتم

همان‌طور که مشاهده می‌کنید پس از شروع الگوریتم در مراحل ۲، ۳ و ۴ سه عدد از ورودی دریافت شده و در متغیرهای X، Y و Z ذخیره می‌شوند؛ در مرحله ۵ این سه متغیر با هم جمع شده و حاصل جمع در متغیر SUM ذخیره می‌شود، سپس در مرحله ۶ حاصل جمع بر عدد ۳ تقسیم می‌شود و میانگین به دست آمده در متغیر AVE ذخیره و در مرحله ۷ این مقدار روی صفحه، نمایش داده می‌شود و در مرحله ۸ الگوریتم پایان می‌یابد.

البته تا جایی که به الگوریتم صدمه‌ای وارد نشود، می‌توانید الگوریتم را خلاصه‌تر و ساده‌تر کنید؛ به عنوان مثال الگوریتم مثال ۲ را می‌توانید به صورت زیر تغییر دهید و سه عدد را در یک مرحله دریافت کرده و برای محاسبه معدل سه عدد مرحله ۵ و ۶ را با هم ترکیب کنید.

توضیح	
۱- شروع	شروع الگوریتم
۲- X و Y و Z را دریافت کن	ورود داده
۳- $AVE \leftarrow \frac{X + Y + Z}{۳}$	محاسبه میانگین
۴- AVE را نمایش بده	نمایش خروجی
۵- پایان	پایان الگوریتم

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۳۱

د ... دستورالعمل‌های شرطی

گاهی اوقات لازم است با مقایسه مقادیر داده‌های ورودی، اطلاعات خروجی، متغیرها و غیره عملیاتی را در الگوریتم هدایت کرده و دستورالعمل‌های خاصی را اجرا کنید یا روند اجرای الگوریتم را با اتخاذ تصمیمات مناسبی کنترل نمایید. در این صورت از دستورات شرطی استفاده کنید؛ شکل کلی یک دستورالعمل شرطی به یکی از این حالت‌هاست:

دستورات (Then) شرط (ها) (IF)

دستورات (Then) شرط (ها) (IF) دستورات (Else) دستورالعمل (ات)

در حالت اول، ابتدا شرط یا شرط‌های آرایه شده بررسی می‌شوند و در صورتی که نتیجه بررسی درست باشد، دستور یا دستورات پس از Then اجرا می‌شوند. در غیراین صورت (نادرست بودن شرط بررسی شده) دستورالعملی که پس از دستورالعمل شرطی قرار گرفته، اجرا خواهد شد بدون آن که دستور یا دستورات پس از Then اجرا شود.

در حالت دوم، شکل کامل‌تری از دستورالعمل شرطی را ملاحظه می‌کنید که در آن ابتدا شرط یا شرط‌ها مورد بررسی قرار می‌گیرند؛ اگر نتیجه ارزیابی آن‌ها درست باشد، دستورات موجود در بخش Then اجرا می‌شوند، سپس دستورالعملی که پس از دستورالعمل شرطی قرار گرفته است، اجرا می‌شود اما اگر نتیجه ارزیابی شرط یا شرط‌ها نادرست باشد بدون آن که دستورات بخش Then اجرا شوند، دستورات موجود در بخش Else اجرا خواهند شد سپس دستورالعملی که پس از دستور شرطی قرار دارد، اجرا می‌شود.

عملگرهای مقایسه‌ای

همان‌طور که گفته شد گاهی لازم است تا از دستورالعمل‌های شرطی در الگوریتم‌ها استفاده کنید. این‌گونه دستورالعمل‌ها، عملیاتی را براساس نتیجه ارزیابی شرط یا شرط‌هایی که پس از عبارت اگر قرار داده می‌شوند، اجرا کرده یا از اجرای آن‌ها جلوگیری می‌کنند. برای طراحی و ایجاد شرط‌های موردنظر، از عملگرهای مقایسه‌ای استفاده می‌شود؛ قبلاً از این‌گونه عملگرها در مباحث ریاضی استفاده کرده‌اید.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳۰۲۲/۲۷	شماره شناسایی: ۳۰۲۲/۲۷-۲	شماره شناسایی: ۳۰۲۲/۲۷-۲۱

جدول ۱-۲

عملگر	عنوان	مثال	نتیجه
=	تساوی	$4=4$	درست
>=	بزرگ‌تر یا مساوی	$5>=2$	درست
<=	کوچک‌تر یا مساوی	$100<=15$	نادرست
>	بزرگ‌تر	$29>74$	نادرست
<	کوچک‌تر	$7<3$	نادرست
<>	نامساوی	$15<>8$	درست

 **نکته:** در صورت نیاز، می‌توان یک دستورالعمل شرطی را در بخش دستورات بعد از *Then* یا *Else* دستورالعمل شرطی دیگری قرار داد؛ در این حالت به مجموعه دستورالعمل‌های شرطی، دستورالعمل‌های شرطی تو در تو نیز می‌گویند.

مثال ۳: الگوریتمی بنویسید که سه عدد دلخواه را دریافت کرده و بزرگ‌ترین مقدار را در بین آن‌ها تعیین کند و نمایش دهد.

در این مسأله سه داده ورودی وجود دارند که آن‌ها را اعداد A، B و C در نظر می‌گیریم، برای پیدا کردن بزرگ‌ترین مقدار در بین این سه عدد می‌توانیم از روش مقایسه استفاده کنیم؛ به این منظور ابتدا فرض می‌کنیم مقدار ذخیره شده در متغیر A بزرگ‌ترین مقدار است و مقدار آن را در متغیر دیگری به نام MAX ذخیره می‌کنیم، سپس متغیر B را با MAX مقایسه کرده و در صورت بزرگ‌تر بودن مقدار B از MAX، متغیر B را در MAX ذخیره می‌کنیم، اگر به همین ترتیب این عملیات را برای متغیر C نیز انجام دهیم، در پایان بزرگ‌ترین عدد در متغیر MAX ذخیره خواهد شد. با توجه به مطالب گفته شده الگوریتم موردنظر به صورت زیر خواهد بود:

توضیح	
شروع الگوریتم	۱- شروع
ورود داده	۲- اعداد A، B و C را دریافت کن
مقدار متغیر A در MAX ذخیره می‌شود.	۳- $MAX \leftarrow A$
در صورتی که مقدار B از MAX بزرگ‌تر باشد مقدار B را در MAX ذخیره می‌کند.	۴- اگر $B > MAX$ آن‌گاه $MAX \leftarrow B$
در صورتی که مقدار C از MAX بزرگ‌تر باشد مقدار C را در MAX ذخیره می‌کند.	۵- اگر $C > MAX$ آن‌گاه $MAX \leftarrow C$

استاندارد مهارت: رابطه کار درجه ۱	بیمه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

توضیح	
۶- MAX را نمایش بده	نمایش خروجی
۷- پایان	پایان الگوریتم

در این جا لازم است شما را با موضوع دیگری آشنا کنیم و آن اجرا و آزمایش الگوریتم (Trace) به وسیله طراح الگوریتم است. یکی از مسائل مهم در طراحی یک الگوریتم، اطمینان از صحت عملکرد آن است، پس باید آن را اجرا کرده و به ازای مقادیر مختلف ورودی، مورد آزمایش قرار داد. برای آشنایی با روش انجام این کار الگوریتمی را که در مثال ۳ طراحی شده است، آزمایش می‌کنیم. ابتدا جدولی را برای بررسی عملکرد الگوریتم به صورت زیر ایجاد کنید:

جدول ۱-۳

خروجی	MAX	C	B	A

اکنون الگوریتم را با اجرای دستورالعمل شماره ۱ آغاز و سپس مرحله ۲ را اجرا کنید، اعداد ۴، ۱۵ و ۲ را به عنوان اعداد موردنظر وارد الگوریتم نمایید و این مقادیر را در جدول مربوطه بنویسید.

جدول ۱-۴

خروجی	MAX	C	B	A
		۲	۱۵	۴

اکنون مرحله ۳ را اجرا کرده و مقدار ذخیره شده در A را در متغیر MAX کپی کنید؛ بنابراین جدول به صورت زیر درمی‌آید:

جدول ۱-۵

خروجی	MAX	C	B	A
	۴	۲	۱۵	۴

در مرحله ۴ دستورالعمل شرطی اول اجرا شده و مقدار متغیر B با MAX مقایسه می‌شود. اگر به جدول مربوطه مراجعه و مقادیر این دو متغیر را ملاحظه کنید، مقایسه $15 > 4$ نتیجه درست را در پی خواهد داشت، بنابراین دستور پس از آن گاه یعنی $B \leftarrow MAX$ اجرا شده و در نتیجه جدول ۱-۶ حاصل می‌شود.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی الگوریتم و فلوجارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

جدول ۱-۶

خروجی	MAX	C	B	A
	۳	۲	۱۵	۴
	۱۵			

– در مرحله ۵، دستورالعمل شرطی دوم اجرا می‌شود و به همان ترتیب مقایسه $C > MAX$ یا به عبارت دیگر $۲ > ۱۵$ انجام می‌شود که نتیجه نادرست را در پی دارد، بنابراین بدون آن که دستور $MAX \leftarrow C$ اجرا شود به مرحله ۶ می‌رسیم و تغییری نیز در جدول ایجاد نمی‌شود.

– در مرحله ۶ مقدار متغیر MAX در خروجی نمایش داده می‌شود و جدول حافظه به صورت زیر در می‌آید:

جدول ۱-۷

خروجی	MAX	C	B	A
	۳	۲	۱۵	۴
۱۵	۱۵			

– با اجرای مرحله ۷ اجرای الگوریتم پایان می‌یابد. اکنون می‌توانید با مقایسه خروجی الگوریتم مزبور با محاسباتی که انجام می‌دهید از صحت عملکرد الگوریتم اطمینان حاصل کنید.

تمرین: الگوریتمی بنویسید که چهار عدد دلخواه را دریافت کرده و کوچک‌ترین و بزرگ‌ترین مقدار در بین آن‌ها را به دست آورد، سپس آن را برای اعداد مختلف اجرا و آزمایش کنید.

مثال ۴: الگوریتمی بنویسید که دو عدد را دریافت کرده و بخش‌پذیری عدد اول را بر عدد دوم بررسی کند.

برای این کار لازم است ابتدا دو عدد را دریافت کرده و سپس با استفاده از عملگر Mod به همراه یک دستورالعمل شرطی، بخش‌پذیری دو عدد را بررسی کنید؛ به این منظور پس از دریافت دو عدد و ذخیره آن‌ها در متغیرهای M و N، در مرحله ۳ باقیمانده تقسیم M بر N بررسی شده و در صورتی که این مقدار برابر با صفر باشد به معنی بخش‌پذیری M بر N خواهد بود و در نتیجه پیام "عدد اول بر عدد دوم بخش پذیر است." نمایش داده می‌شود، در غیر این صورت پیام "عدد اول بر عدد دوم بخش پذیر نیست." نمایش داده خواهد شد و الگوریتم پایان می‌یابد.

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

اکنون الگوریتم موردنظر به صورت زیر خواهد بود:

۱- شروع

۲- M و N را دریافت کن

۳- اگر $M \bmod N = 0$ آن‌گاه "عدد اول بر عدد دوم بخش‌پذیر است" را نمایش بده در غیراین صورت "عدد اول بر عدد دوم بخش‌پذیر نیست" را نمایش بده

۴- پایان

تمرین: الگوریتمی بنویسید که زوج یا فرد بودن هر عدد دلخواه را معین کند.

عملگرهای منطقی

علاوه بر عملگرهای ریاضی و مقایسه‌ای، نوع دیگری از عملگرها به نام عملگرهای منطقی نیز وجود دارند که برای ترکیب چند شرط و مقایسه آن‌ها با هم مورد استفاده قرار می‌گیرند. این عملگرها در جدول ۱-۸ ارائه شده‌اند.

جدول ۱-۸

عملگر	مفهوم
AND	«و» منطقی
OR	«یا» منطقی
NOT	نقیض

وقتی دو یا چند شرط با عملگر منطقی AND با هم ترکیب می‌شوند، نتیجه مقایسه، زمانی True خواهد بود که نتیجه تمام شرط‌ها و مقایسه‌ها True باشند؛ بنابراین اگر عبارات مقایسه‌ای A و B را در نظر بگیرید، جدول درستی عملگر منطقی AND مطابق جدول ۱-۹ خواهد بود.

جدول ۱-۹

A	B	A AND B
T	F	F
F	T	F
F	F	F
T	T	T

در صورتی که دو یا چند شرط یا مقایسه را با عملگر منطقی OR با هم ترکیب کنید، نتیجه مقایسه زمانی False (نادرست) خواهد بود که نتیجه تمام شرط‌ها و مقایسه‌ها False (نادرست) باشد و

استاندارد مهارت، رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

در سایر حالات، نتیجه True (درست) خواهد بود؛ بنابراین جدول درستی عملگر منطقی OR مطابق جدول ۱-۱۰ است.

جدول ۱-۱۰

A	B	A OR B
T	F	T
F	T	T
F	F	F
T	T	T

مثال ۵: الگوریتمی بنویسید که حقوق و دستمزد کارگران یک کارخانه را مطابق شرایط زیر محاسبه کند:

الف- برای کارگرانی که دارای بیش از ۱۰ سال سابقه کار بوده و سن آن‌ها ۳۰ سال یا بیشتر است، حقوق خالص به علاوه ۱۰ درصد حقوق به صورت پاداش پرداخت شود.

ب- برای کارگرانی که دارای بیش از ۱۵ سال سابقه کار بوده و سن آن‌ها ۴۰ سال یا بیشتر است، حقوق خالص به علاوه ۲۰ درصد حقوق به صورت پاداش پرداخت شود.

ج- برای کارگرانی که دارای بیش از ۲۵ سال سابقه کار بوده یا سن آن‌ها ۵۰ سال یا بیشتر است، حقوق خالص به علاوه ۲۵ درصد حقوق به صورت پاداش پرداخت شود.

د- برای کارگرانی که دارای سابقه کاری به مدت ۱۰ سال یا کمتر بوده یا سن آن‌ها ۳۰ سال یا کمتر است، حقوق خالص به علاوه ۲ درصد حقوق به صورت پاداش پرداخت شود.

همان‌طور که ملاحظه می‌کنید، برای طراحی الگوریتمی که بتواند حقوق کارگران را محاسبه کند شرایط مختلفی در نظر گرفته شده است. برای حل این مثال دو روش وجود دارد: روش اول استفاده از دستورالعمل‌های شرطی تودرتو و روش دوم استفاده از عملگرهای منطقی است.

در روش اول لازم است تا برای بندهای الف و ب از یک دستورالعمل شرطی در بخش آن‌گاه دستورالعمل شرطی دیگر استفاده شود تا هر دو شرط موردنظر در این شرایط مورد ارزیابی قرار گیرند اما برای شرایط ج و د می‌توان از دستورالعمل‌های شرطی جداگانه استفاده کرد و چون برقراری یکی از شرط‌ها در این شرایط برای محاسبه حقوق براساس فرمول مربوطه کافی است، با ارایه این توضیحات می‌توانید الگوریتم زیر را بنویسید:

در این الگوریتم Salary به عنوان میزان حقوق، Age به عنوان سن کارگری و W به عنوان سابقه کار کارگر در نظر گرفته شده‌اند.

استاندارد مهارت: رایانه کار درجه ۱	بیمه مهرنی: الگوریتم و فلوجارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۳۱

۱- شروع

۲- Salary و Age و W را دریافت کن

۳- $\leftarrow \text{Payment}$ (صفر به عنوان مقدار اولیه در Payment ذخیره می‌شود).

۴- اگر $W > 10$ آن‌گاه اگر $\text{Age} \geq 30$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/1 \times \text{Salary})$

۵- اگر $W > 15$ آن‌گاه اگر $\text{Age} \geq 40$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/2 \times \text{Salary})$

۶- اگر $W > 25$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/25 \times \text{Salary})$

۷- اگر $\text{Age} \geq 50$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/25 \times \text{Salary})$

۸- اگر $W \leq 10$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/02 \times \text{Salary})$

۹- اگر $\text{Age} \leq 30$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/02 \times \text{Salary})$

۱۰- Payment را نمایش بده

۱۱- پایان

البته الگوریتم فوق را می‌توانید به صورت ساده‌تر و جامع‌تری نیز بنویسید، برای این کار لازم است تا از عملگرهای منطقی استفاده کرده و شرط‌های موجود در دستورالعمل‌های شرطی تودرتو را با هم ترکیب کنید، بنابراین می‌توانید الگوریتم فوق را به صورت زیر بنویسید:

۱- شروع

۲- Salary, Age و W را دریافت کن

۳- $\leftarrow \text{Payment}$

۴- اگر $W > 10$ و $\text{Age} \geq 30$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/1 \times \text{Salary})$

۵- اگر $W > 15$ و $\text{Age} \geq 40$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/2 \times \text{Salary})$

۶- اگر $W > 25$ یا $\text{Age} \geq 50$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/25 \times \text{Salary})$

۷- اگر $W \leq 10$ یا $\text{Age} \leq 30$ آن‌گاه $\text{Payment} \leftarrow \text{Salary} + (0/02 \times \text{Salary})$

۸- Payment را نمایش بده

۹- پایان

تمرین: الگوریتمی بنویسید تا نمرات ۴ درس یک دانش آموز را دریافت کند و با توجه به میانگین نمرات، رتبه وی را براساس شرایطی که در ادامه می‌آید، تعیین کند:

۱- حقوق

۲- سن

۳- حقوق پرداختی

استاندارد مهارت: رانندگی کار درجه ۱	پیمانه مهارتی الگوریتم و فلوچارت	واحد کار حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

- الف- در صورتی که میانگین نمرات بیشتر از ۱۸ باشد، رتبه ممتاز برای وی اعلام شود.
- ب- در صورتی که میانگین نمرات بین ۱۶ تا ۱۸ و نمره ریاضی بزرگ‌تر از ۱۷ باشد، رتبه خوب برای وی اعلام شود.
- ج- در صورتی که میانگین نمرات بین ۱۴ تا ۱۶ و نمره ریاضی یا فیزیک بزرگ‌تر از ۱۵ باشد، رتبه متوسط برای وی اعلام شود.
- د- در صورتی که میانگین نمرات بین ۱۲ تا ۱۴ و نمره ریاضی یا فیزیک یا زبان بزرگ‌تر از ۱۴ باشد، رتبه ضعیف برای وی اعلام شود.
- ه- در صورتی که میانگین نمرات کمتر از ۱۲ و نمره ریاضی و فیزیک کوچک‌تر از ۱۲ باشد، رتبه خیلی ضعیف برای وی اعلام شود.

ه- دستورالعمل‌های تکرار (حلقه‌ها)

استفاده از دستورالعمل‌هایی که تاکنون فراگرفته‌اید در حل بعضی از مسائل کافی نیست و بعضی از اوقات لازم است تا برخی از دستورالعمل‌ها را به دفعات تکرار نمایید؛ در این موارد از دستورالعمل تکرار یا همان حلقه‌ها استفاده کنید. دستورالعمل حلقه از اجزای مختلفی تشکیل می‌شود که عبارتند از:

شمارنده حلقه: یک متغیر عددی است که تعداد دفعات تکرار دستورالعمل‌ها را در حلقه کنترل می‌کند. مقدار شمارنده در هر بار اجرای حلقه افزایش یا کاهش می‌یابد.

مقدار اولیه: مقدار اولیه حلقه قبل از شروع حلقه تعیین می‌شود و به وسیله آن می‌توان مقدار اولیه را برای شمارنده حلقه تعیین کرد.

شرط حلقه: برای کنترل تعداد دفعات تکرار حلقه، باید از یک شرط استفاده کرد. شرط موجود در حلقه، نقطه پایان تکرار دستورالعمل‌ها را در حلقه مشخص می‌کند و باید به گونه‌ای تنظیم شود تا از ایجاد حلقه نامحدود جلوگیری کند. برای ایجاد شرط در یک حلقه می‌توان از دستورالعمل‌های شرطی استفاده کرد.

دستورات حلقه: بخش دیگر در حلقه، دستورالعمل‌هایی هستند که در داخل حلقه تکرار می‌شوند. این دستورالعمل‌ها با توجه به نیاز مسأله انتخاب می‌شوند.

مثال ۶: الگوریتمی بنویسید که اعداد طبیعی کوچک‌تر از ۵۰ را نمایش دهد.

برای طراحی این الگوریتم لازم است به مفهوم آن توجه داشته باشید. ظاهراً این مسأله داده ورودی ندارد ولی با کمی توجه می‌توان دریافت که اولین مقدار برای شروع اعداد طبیعی عدد ۱ است و در این مجموعه اعداد، هر عدد با عدد بعدی یک واحد اختلاف دارد، بنابراین به راحتی می‌توان با

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

استفاده از یک حلقه به جای استفاده از چندین دستورالعمل، از یک دستورالعمل به صورت مشترک استفاده کرد. به این منظور کافی است یک متغیر مانند N را در نظر گرفته و مقدار ۱ را در آن ذخیره کرد و سپس آن را نمایش داد؛ در مرحله بعد یک واحد به آن اضافه کرده $(N+1)$ و سپس مجدداً آن را (N) نمایش داد. با تکرار این فرآیند اعداد طبیعی ۱ به بعد را نمایش دهید و برای آن که از محدوده عدد ۵۰ خارج نشوید، می‌توانید از یک دستورالعمل شرطی استفاده کنید.

با توضیحاتی که داده شد الگوریتم موردنظر به صورت زیر به دست می‌آید:

توضیح	
۱- شروع	
۲- $N \leftarrow 1$	تنظیم مقدار اولیه برای شمارنده حلقه
۳- N را نمایش بده	نمایش خروجی
۴- $N \leftarrow N+1$	افزایش شمارنده حلقه
۵- اگر $N < 50$ آن‌گاه به مرحله ۳ برو	شرط حلقه
۶- پایان	

همان‌طور که می‌بینید، با شروع الگوریتم ابتدا عدد ۱ در متغیر N ذخیره شده، سپس این مقدار در مرحله ۳ روی صفحه نمایش نشان داده می‌شود. در مرحله بعد دستور $N \leftarrow N+1$ سبب افزایش مقدار N به میزان ۱ واحد خواهد شد زیرا ابتدا عبارت سمت راست انجام می‌شود و به مقدار فعلی متغیر N یک واحد اضافه کرده و نتیجه را مجدداً در همان محل از حافظه یعنی متغیر N ذخیره می‌کند؛ این کار باعث می‌شود تا مقدار جدید جایگزین مقدار قبلی در متغیر N شود. به این صورت اعداد طبیعی کوچک‌تر از ۵۰ یکی یکی تولید می‌شوند.

در مرحله ۵ با استفاده از یک دستورالعمل شرطی مقدار N بررسی می‌شود؛ در صورتی که مقدار آن کوچک‌تر از ۵۰ باشد دستور بعد از آن‌گاه اجرای الگوریتم را به مرحله ۳ منتقل می‌کند و دستورات مراحل ۳ و ۴ آن‌قدر تکرار می‌شوند تا مقدار N برابر ۵۰ شود، در این زمان نتیجه ارزیابی شرط $N < 50$ نادرست خواهد بود و در نتیجه اجرای حلقه ایجاد شده خاتمه یافته، مرحله ۶ اجرا شده و سبب خاتمه اجرای الگوریتم می‌شود. اگر الگوریتم را اجرا و آزمایش کنید، جدول بررسی عملکرد آن به صورت جدول ۱۱-۱ خواهد بود.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

جدول ۱۱-۱

خروجی	N
۱	X
۲	X
۳	X
۴۹	۴۹
	۵۰

مثال ۷: الگوریتمی بنویسید که اعداد زوج کوچک‌تر یا مساوی عدد طبیعی و دلخواه N را نمایش دهد. این الگوریتم مشابه الگوریتم قبلی است با این تفاوت که مقدار شمارنده از ۲ شروع شده و هر بار به میزان ۲ واحد افزایش می‌یابد.

توضیح

۱- شروع
۲- N را دریافت کن
۳- اگر $N = 1$ آن‌گاه پایان
۴- $K \leftarrow 2$
۵- K را نمایش بده
۶- $K \leftarrow K + 2$
۷- اگر $K \leq N$ آن‌گاه برو به مرحله ۵
۸- پایان

تمرین: الگوریتمی بنویسید که مضارب کوچک‌تر از ۱۰۰ عدد ۳ را نمایش دهد.

مثال ۸: الگوریتمی بنویسید که مجموع و تعداد اعداد طبیعی موجود بین اعداد طبیعی دلخواه M و N را نمایش دهد (با فرض این که M کوچک‌تر از N باشد).

۱- شروع

۲- M و N را دریافت کن

۳- $SUM \leftarrow 0$ و $NO \leftarrow 0$

۴- $K \leftarrow M + 1$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

$$5-SUM \leftarrow SUM + K$$

$$6-NO \leftarrow NO + 1$$

$$7-K \leftarrow K + 1$$

۸- اگر $K < N$ آن‌گاه برو به مرحله ۵

۹- SUM و NO را نمایش بده

۱۰- پایان

نکته‌ای که باید در این الگوریتم بدان اشاره کنیم نحوه محاسبه مجموع و تعداد اعداد طبیعی است. برای این کار از متغیر SUM برای محاسبه مجموع اعداد طبیعی و از متغیر NO برای شمارش تعداد اعداد بین M و N استفاده می‌شود. متغیر K برای ایجاد اعداد طبیعی بین M و N به کار می‌رود. در صورت اجرای این الگوریتم به ازای $M=3$ و $N=9$ ، جدول بررسی عملکرد به صورت زیر خواهد بود:

جدول ۱۲-۱

خروجی	No	SUM	K	N	M
	۵	۳۰	۹	۹	۳
	۴	۲۹	۸		
	۳	۲۵	۷		
	۲	۲۱	۶		
	۱	۱۵	۵		

تمرین: الگوریتم فوق را به گونه‌ای تنظیم کنید که بدون در نظر گرفتن فرض $M < N$ ، توانایی ارایه پاسخ صحیح را داشته باشد؛ به عبارت دیگر در صورتی که کاربر عدد بزرگ‌تر را برای M و عدد کوچک‌تر را برای N وارد کند، پاسخ الگوریتم درست باشد.

مثال ۹: الگوریتمی بنویسید که عدد طبیعی و دلخواه N را دریافت کرده و مقدار $N!$ (فاکتوریل) را محاسبه کند و نمایش دهد.

تعریف ریاضی فاکتوریل که با علامت $N!$ نشان داده می‌شود به معنی حاصل ضرب عدد N در اعداد طبیعی کوچک‌تر از آن است. به عنوان مثال مقدار $5!$ به صورت زیر محاسبه می‌شود:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

نکته: مقدار $1! = 1$ و $0! = 1$ است.



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

در این مسأله داده ورودی عدد N و اطلاعات خروجی مقدار فاکتوریل عدد N است؛ محاسبات نیز برطبق مثال کاملاً واضح و معین است اما برای به دست آوردن مقدار فاکتوریل یک عدد همان‌طور که از رابطه اعداد در مثال آرایه شده مشخص است باید اعداد طبیعی کوچک‌تر یا مساوی عدد N را در هم ضرب کرد، بنابراین استفاده از یک حلقه برای انجام این کار الزامی است؛ به علاوه برای ایجاد اعداد طبیعی کوچک‌تر از N باید یک متغیر (K) اختیار کرده و مقدار اولیه آن را برابر N قرار داد، هم‌چنین برای به دست آوردن مقدار فاکتوریل می‌توانید از یک متغیر دیگر (Fact) استفاده کرده و مقدار اولیه آن را روی ۱ تنظیم کنید و با استفاده از دستورالعمل $Fact \leftarrow Fact \times K$ مقدار فاکتوریل را محاسبه کنید زیرا اگر مقدار Fact برابر صفر باشد هر عددی که در آن ضرب شود، از بین می‌رود ولی اگر مقدار Fact برابر یک باشد هر عددی که در آن ضرب شود، در عدد قبلی ضرب خواهد شد. با این توضیحات الگوریتم موردنظر به صورت زیر خواهد بود:

- ۱- شروع
- ۲- N را دریافت کن
- ۳- $K \leftarrow N$ و $Fact \leftarrow 1$
- ۴- $Fact \leftarrow Fact \times K$
- ۵- $K \leftarrow K - 1$
- ۶- اگر $K \geq 1$ آن‌گاه برو به مرحله ۴
- ۷- Fact را نمایش بده
- ۸- پایان

تمرین: الگوریتم فوق را به گونه‌ای تغییر دهید که مقدار فاکتوریل را با استفاده از یک حلقه با شمارنده افزایشی محاسبه کند.

مثال ۱۰: الگوریتمی بنویسید که مقسوم‌علیه‌های عدد طبیعی و دلخواه M را محاسبه کرده و نمایش دهد.

به این منظور باید از یک حلقه استفاده کنید و متغیر M را هر بار بر یک متغیر K که به عنوان مقسوم‌علیه در نظر گرفته می‌شود و مقدار اولیه آن ۱ است، تقسیم نمایید سپس باقیمانده این تقسیم را بررسی کنید. در صورتی که مقدار باقیمانده مساوی صفر باشد به این معنی است که متغیر K می‌تواند مقسوم‌علیه متغیر M باشد، بنابراین متغیر K نمایش داده می‌شود. برای محاسبه باقیمانده تقسیم صحیح می‌توانید از عملگر Mod استفاده کنید. به عنوان نمونه به این مثال توجه کنید:

$$\begin{array}{r} 17 \overline{) 5} \\ 15 \quad 3 \end{array}$$

باقیمانده تقسیم صحیح ۳ ← ۲

واحد کار حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوجارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۲-۴۲/۲۷-۲۱	شماره شناسایی: ۲-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

$$17 \text{ Mod } 5 = 2$$

اکنون الگوریتم موردنظر به صورت زیر خواهد بود:

- ۱- شروع
- ۲- M را دریافت کن
- ۳- $K \leftarrow 1$
- ۴- اگر $M \text{ Mod } K = 0$ آن گاه K را نمایش بده
- ۵- $K \leftarrow K + 1$
- ۶- اگر $K \leq M$ آن گاه برو به مرحله ۴
- ۷- پایان

مثال ۱۱: الگوریتمی بنویسید که مجموع ارقام هر عدد طبیعی و دلخواه را محاسبه کرده و نمایش دهد. برای طراحی الگوریتم فوق و محاسبه مجموع ارقام یک عدد، ابتدا باید ارقام عدد را یکی یکی از آن جدا کنید؛ به این منظور می‌توانید از عملگر Mod و محاسبه باقیمانده تقسیم عدد مربوطه بر عدد ۱۰ استفاده نمایید، سپس خارج قسمت عدد مربوطه را که بر ۱۰ تقسیم می‌شود، محاسبه کنید. از این عدد می‌توانید برای به دست آوردن رقم بعدی در عدد اصلی استفاده نمایید. این عملیات می‌تواند در یک حلقه تا رسیدن به خارج قسمت صفر ادامه یابد. برای محاسبه خارج قسمت تقسیم صحیح نیز می‌توانید از عملگر $\backslash$ استفاده کنید؛ به عنوان نمونه به این مثال توجه کنید:

$$\begin{array}{r}
 17 \overline{) 5} \\
 \underline{15} \quad \textcircled{3} \quad \leftarrow \text{خارج قسمت تقسیم صحیح} \\
 2 \quad \leftarrow \text{باقیمانده تقسیم صحیح}
 \end{array}$$

$17 \backslash 5 = 3$

با توجه به مطالب ارایه شده، اگر بخواهیم ارقام یک عدد را از آن جدا کنیم روش ریاضی این عملیات به صورت زیر خواهد بود:

به عنوان مثال اگر عدد مورد نظر ۴۲۵۷ باشد:

$$\begin{array}{r}
 4257 \overline{) 10} \\
 \underline{4250} \quad \textcircled{7} \\
 10
 \end{array}
 \quad
 \begin{array}{r}
 425 \overline{) 10} \\
 \underline{420} \quad \textcircled{5} \\
 10
 \end{array}
 \quad
 \begin{array}{r}
 42 \overline{) 10} \\
 \underline{40} \quad \textcircled{2} \\
 10
 \end{array}
 \quad
 \begin{array}{r}
 4 \overline{) 10} \\
 \underline{0} \quad \textcircled{4} \\
 10
 \end{array}$$

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رابطه کار درجه ۱
شماره شناسایی: ۲۱-۴۲/۲۷-۳	شماره شناسایی: ۲-۴۲/۲۷-۳	شماره شناسایی: ۳-۴۲/۲۷

بنابراین می‌توان الگوریتم زیر را جهت حل این مثال ارایه کرد:

۱- شروع

۲- M را دریافت کن

۳- $SUM \leftarrow 0$

۴- $Digit \leftarrow M \bmod 10$

۵- $SUM \leftarrow SUM + Digit$

۶- $M \leftarrow M \div 10$

۷- اگر $M > 0$ آن‌گاه برو به مرحله ۴

۸- SUM را نمایش بده

۹- پایان

همان‌طور که مشاهده می‌کنید پس از دریافت عدد موردنظر و صفر کردن متغیر SUM ، در مرحله چهارم اولین رقم از عدد M جدا شده و در متغیر $Digit$ ذخیره می‌شود، سپس در مرحله بعد متغیر $Digit$ به مقدار SUM اضافه می‌شود تا جمع ارقام محاسبه شود. در مرحله ۶ با استفاده از عملگر $\div$ رقم سمت راست از عدد حذف می‌شود و مابقی ارقام به عنوان عدد جدید در متغیر M قرار می‌گیرند، سپس با استفاده از یک دستورالعمل شرطی مقدار M بررسی می‌شود و در صورتی که مقدار آن مخالف صفر باشد، مراحل ۴ تا ۷ مجدداً تکرار می‌شوند. زمانی که مقدار M صفر شود به معنی تمام شدن ارقام عدد M خواهد بود در نتیجه مرحله ۸ مجموع ارقام (SUM) را نمایش خواهد داد و الگوریتم در مرحله ۹ پایان می‌یابد.

تمرین: الگوریتم را به ازای $M=74825$ آزمایش کنید.

تمرین: الگوریتمی بنویسید که ارقامی از عدد طبیعی و دلخواه N را نمایش دهد که بر ۵ قابل تقسیم باشند.

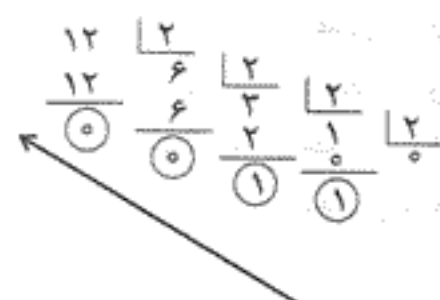
تبدیل اعداد از مبنای ۱۰ به مبنای ۲:

همان‌طور که می‌دانید عددنویسی در ریاضیات در مبناهای مختلفی وجود دارد و می‌توانید در هر مبنایی اعداد خود را بنویسید. به طور معمول از مبنای ۱۰ یا دسیمال استفاده می‌کنید و به این منظور ارقام ۰ تا ۹ را (۱۰ رقم) به کار می‌گیرید. مبناهای دیگری نیز وجود دارند، به عنوان مثال می‌توان عددنویسی در مبنای ۲ را نام برد.

هم‌چنین می‌دانید که مبنای پردازش و هرگونه عملیات در کامپیوتر براساس مبنای ۲ یا همان باینری (دودویی) است، البته مبناهای ۱۶ نیز در کامپیوتر مورد استفاده قرار می‌گیرند. برای تبدیل

استاندارد مهارت: رابطه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

اعداد از مبنای ۱۰ به مبنای ۲ می‌توانید از روش تقسیم استفاده کنید. به عنوان مثال فرض کنید می‌خواهیم عدد ۱۲ را به مبنای ۲ تبدیل کنیم بنابراین مراحل انجام کار به این صورت خواهد بود:



در پایان اعداد را از پایین به بالا و از چپ به راست به صورت زیر بنویسید:

$$(12)_{10} = (1100)_2$$

تمرین: اعداد ۴۳ و ۱۲۲ را به مبنای ۲ بنویسید.

مثال ۱۲: الگوریتمی بنویسید که یک عدد طبیعی در مبنای ۱۰ را دریافت کند و معادل آن را در مبنای ۲ محاسبه کرده و نمایش دهد.

۱- شروع

۲- M را دریافت کن

۳- $binary \leftarrow 0$ و $P \leftarrow 0$

۴- $digit \leftarrow M \text{ Mod } 2$

۵- $binary \leftarrow binary + digit \times 10^P$

۶- $P \leftarrow P + 1$

۷- $M \leftarrow M \setminus 2$

۸- اگر $M > 0$ آن‌گاه به مرحله ۴ برو

۹- $binary$ را نمایش بده

۱۰- پایان

تمرین: الگوریتم فوق را برای $M=54$ آزمایش کنید.

تبدیل اعداد از مبنای ۲ به مبنای ۱۰:

مانند تبدیل اعداد از مبنای ۱۰ به ۲ برای تبدیل اعداد از مبنای ۲ به ۱۰ نیز راه حل ریاضی به شما کمک می‌کند. همان‌طور که می‌دانید در مبنای ۱۰، ارزش هر رقم به مقدار رقم و ارزش مکانی که

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۲-۴۲/۲۷-۲۱	شماره شناسایی: ۲-۴۲/۲۷-۲	شماره شناسایی: ۲-۴۲/۲۷

$$M \leftarrow M \setminus 16$$

۱۳- اگر $M \neq 0$ آن‌گاه برو به مرحله ۳

۱۴- Hex را نمایش بده

۱۵- پایان

همان‌طور که مشاهده می‌کنید با استفاده از دستورالعمل‌های شرطی متعدد ارقام بزرگ‌تر از ۱۰ به حروف معادل خود تبدیل می‌شوند و سپس به متغیر Hex که به عنوان معادل در مبنای ۱۶ است، اضافه شده و در پایان عدد معادل در مبنای ۱۶ نمایش داده می‌شود.

تبدیل اعداد از مبنای ۱۶ به مبنای ۱۰:

برای تبدیل اعداد از مبنای ۱۶ به مبنای ۱۰ از همان الگوی تبدیل اعداد از مبنای ۲ به ۱۰ استفاده کنید، با این تفاوت که از توان‌های متوالی عدد ۱۶ استفاده می‌شود، به عنوان مثال برای تبدیل اعداد (۳BY) به مبنای ۱۰ به این صورت عمل کنید:

$$\begin{array}{ccc} 3 & B & Y \\ \downarrow & \downarrow & \downarrow \\ 16^2 & 16^1 & 16^0 \end{array}$$

سپس هر رقم را در ارزش مکانی آن ضرب کرده و حاصل ضرب‌ها را با هم جمع کنید:

$$3 \times 16^2 + 11 \times 16^1 + 25 \times 16^0 = 951$$

مثال ۱۵: الگوریتمی بنویسید که یک عدد دلخواه را در مبنای ۱۶ دریافت کرده و معادل آن را در مبنای ۱۰ محاسبه نموده و نمایش دهد.

۱- شروع

۲- I ← ۱ و Hex ← ""

۳- Hex را دریافت کن

۴- I امین حرف را از سمت راست متغیر Hex جدا کرده و در متغیر S قرار بده

۵- اگر $S = "A"$ آن‌گاه $digit \leftarrow 10$

۶- اگر $S = "B"$ آن‌گاه $digit \leftarrow 11$

۷- اگر $S = "C"$ آن‌گاه $digit \leftarrow 12$

۸- اگر $S = "D"$ آن‌گاه $digit \leftarrow 13$

۹- اگر $S = "E"$ آن‌گاه $digit \leftarrow 14$

۱۰- اگر $S = "F"$ آن‌گاه $digit \leftarrow 15$

۱۱- اگر $S = "9"$ آن‌گاه $digit \leftarrow 9$

۱۲- اگر $S = "8"$ آن‌گاه $digit \leftarrow 8$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

۱۰ → A

۱۱ → B

۱۲ → C

۱۳ → D

۱۴ → E

۱۵ → F

برای تبدیل اعداد از مبنای ۱۰ به مبنای ۱۶ روش کار مانند حالت تبدیل اعداد از مبنای ۱۰ به مبنای ۲ است با این تفاوت که عدد موردنظر بر ۱۶ تقسیم می‌شود. به عنوان مثال معادل عدد ۷۴۲ در مبنای ۱۰، در مبنای ۱۶ به این صورت محاسبه می‌شود:

$$\begin{array}{r}
 742 \div 16 \\
 \underline{736} \\
 6
 \end{array}
 \quad
 \begin{array}{r}
 16 \div 16 \\
 \underline{16} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 16 \div 16 \\
 \underline{16} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 16 \div 16 \\
 \underline{16} \\
 0
 \end{array}$$

$\begin{array}{c} 6 \\ 0 \\ 0 \\ 0 \end{array} \rightarrow E$

پس از انجام تقسیم، باقیمانده را از پایین به بالا و از چپ به راست بنویسید اما توجه داشته باشید که اگر باقیمانده‌هایی که به دست می‌آیند بزرگ‌تر از ۹ باشند، مطابق قرارداد به حروف A تا F تبدیل می‌شوند بنابراین عدد معادل ۷۴۲ در مبنای ۱۶، عدد (۲E۶) خواهد بود.

مثال ۱۴: الگوریتمی بنویسید که یک عدد دلخواه را در مبنای ۱۰ دریافت کرده و معادل آن را در مبنای ۱۶ محاسبه نموده و نمایش دهد.

۱- شروع

۲- M را دریافت کن

۳- $digit \leftarrow M \text{ Mod } 16$

۴- اگر $digit = 10$ آن‌گاه $A \leftarrow S$

۵- اگر $digit = 11$ آن‌گاه $B \leftarrow S$

۶- اگر $digit = 12$ آن‌گاه $C \leftarrow S$

۷- اگر $digit = 13$ آن‌گاه $D \leftarrow S$

۸- اگر $digit = 14$ آن‌گاه $E \leftarrow S$

۹- اگر $digit = 15$ آن‌گاه $F \leftarrow S$

۱۰- اگر $digit < 10$ آن‌گاه $digit \leftarrow S$

۱۱- $Hex \leftarrow Hex + S$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۴۷	شماره شناسایی: ۳-۴۲/۴۷-۲	شماره شناسایی: ۳-۴۲/۴۷-۲۱

در آن قرار گرفته است، بستگی دارد. در این مبنا ارزش مکانی ارقام در یک عدد برحسب توان‌های متوالی عدد ۱۰ یعنی $10^0, 10^1, 10^2$ و الی آخر سنجیده می‌شود؛ این قانون برای تمام مبناها برقرار است، پس اگر بخواهید عددی را از مبنای ۲ به مبنای ۱۰ بپیرید، باید هر رقم را در ارزش مکانی آن براساس توان‌های متوالی عدد ۲ ضرب کنید.

به عنوان مثال عدد $(11010)_2$ را با این روش به مبنای ۱۰ تبدیل می‌کنیم. برای این کار ابتدا ارزش مکانی ارقام را در عدد مورد نظر به صورت زیر تعیین کنید:



سپس هر رقم را در ارزش مکانی آن ضرب کرده و حاصل ضرب‌ها را با هم جمع می‌کنیم.

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 0 + 2 + 0 = 26$$

مثال ۱۳: الگوریتمی بنویسید که هر عدد باینری دلخواه را به عددی در مبنای ۱۰ تبدیل کرده و نمایش دهد.

۱- شروع

۲- B را دریافت کن

۳- $P \leftarrow 0$ و decimal $\leftarrow 0$

۴- $digit \leftarrow B \text{ Mod } 10$

۵- $decimal \leftarrow decimal + digit \times 2^P$

۶- $P \leftarrow P + 1$

۷- $B \leftarrow B \setminus 10$

۸- اگر $B \neq 0$ آن‌گاه به مرحله ۴ برو

۹- decimal را نمایش بده

۱۰- پایان

تبدیل اعداد از مبنای ۱۰ به مبنای ۱۶:

برطبق مطالبی که در مثال‌های قبل گفته شد برای عددنویسی در مبنای n ، به ارقام صفر تا $n-1$ نیاز داریم بنابراین برای عددنویسی در مبنای ۱۶ احتیاج به ارقام صفر تا ۱۵ خواهیم داشت، در نتیجه برای ارقام ۱۰ تا ۱۵ از حروف A تا F استفاده می‌شود؛ بنابراین می‌توان در این مبنا عددی به صورت $(2DAF)_{16}$ داشت.

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

۱۳- اگر $S = "۷"$ آن‌گاه $digit \leftarrow ۷$

۱۴- اگر $S = "۶"$ آن‌گاه $digit \leftarrow ۶$

۱۵- اگر $S = "۵"$ آن‌گاه $digit \leftarrow ۵$

۱۶- اگر $S = "۴"$ آن‌گاه $digit \leftarrow ۴$

۱۷- اگر $S = "۳"$ آن‌گاه $digit \leftarrow ۳$

۱۸- اگر $S = "۲"$ آن‌گاه $digit \leftarrow ۲$

۱۹- اگر $S = "۱"$ آن‌گاه $digit \leftarrow ۱$

۲۰- اگر $S = "۰"$ آن‌گاه $digit \leftarrow ۰$

۲۱- $decimal \leftarrow decimal + digit \times ۱۶^{I-1}$

۲۲- $I \leftarrow I + ۱$

۲۳- I امین حرف را از سمت راست متغیر Hex جدا کرده و در متغیر S قرار بده

۲۴- اگر $S < ">"$ آن‌گاه برو به مرحله ۵

۲۵- decimal را نمایش بده

۲۶- پایان

همان‌طور که در الگوریتم فوق مشاهده می‌کنید ابتدا عدد در مبنای ۱۶ به صورت یک عبارت که می‌تواند ترکیبی از ارقام و حروف باشد، دریافت شده و در متغیر Hex ذخیره می‌شود و سپس در مرحله ۴، اولین حرف یا رقم سمت راست از متغیر Hex جدا شده و در متغیر S ذخیره می‌شود و بعد از آن در مراحل ۵ تا ۲۰ براساس مقدار متغیر S رقم معادل آن در مبنای ۱۰ به دست می‌آید و در مرحله ۲۱ این رقم با توجه به ارزش مکانی آن در توان مربوطه در عدد ۱۶ ضرب و حاصل ضرب به متغیر decimal اضافه می‌شود تا عدد معادل در مبنای ۱۰ ساخته شود. در مراحل ۲۲ الی ۲۴ ارقام بعدی از عدد در مبنای ۱۶ جدا می‌شود و تا زمانی که ارقام آن تمام شود (" $S < ">"$) اجرای الگوریتم به مرحله ۵ منتقل می‌شود و با پایان یافتن ارقام عدد در مبنای ۱۶ مرحله ۲۵ اجرا می‌شود و عدد معادل را در مبنای ۱۰ نمایش می‌دهد و در مرحله ۲۶ الگوریتم پایان می‌یابد.

مثال ۱۶: الگوریتمی بنویسید که حاصل جمع زیر را براساس مقدار x و n محاسبه کرده و نمایش دهد. x یک عدد حقیقی و n یک عدد طبیعی زوج است ($n \geq ۲$). به عنوان مثال اگر $n=۴$ باشد حاصل جمع به صورت $SUM = x^۲ + x^۴$ محاسبه می‌شود.

$$SUM = x^۲ + x^۴ + x^۶ + \dots + x^{2n}$$

۱- شروع

۲- x و n را دریافت کن

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	بیمه مهارتی الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

$$3- \quad I \leftarrow 2 \text{ و } SUM \leftarrow 0$$

$$4- \quad SUM \leftarrow SUM + x^I$$

$$5- \quad I \leftarrow I + 2$$

$$6- \quad \text{اگر } I \leq n \text{ آن‌گاه برو به مرحله ۴}$$

$$7- \quad SUM \text{ را نمایش بده}$$

$$8- \quad \text{پایان}$$

تمرین: الگوریتمی بنویسید که حاصل جمع زیر را محاسبه کند ($n \geq 1$).

$$SUM = \frac{1}{x} + \frac{2}{x^2} + \frac{5}{x^5} + \dots + \frac{(2n-1)}{x^{(2n-1)}}$$

مثال ۱۷: الگوریتمی بنویسید که مجموع عبارت زیر را محاسبه کرده و نمایش دهد (با فرض آن که $n \geq 0$).

$$SUM = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^{(2n+1)}}{(2n+1)!}$$

در بسط فوق، x مقدار زاویه و n تعداد جملات برای محاسبه سینوس است که دقت محاسبه را تعیین می‌کند و مقدار آن بزرگ‌تر یا مساوی صفر است. برای محاسبه این حاصل جمع باید از حلقه‌های تودرتو استفاده کنیم زیرا علاوه بر این که باید برای محاسبه جملات و جمع آن‌ها با هم از حلقه استفاده شود، در داخل این حلقه باید مقدار فاکتوریل نیز برای هر جمله محاسبه شده و همان‌طور که قبلاً مشاهده کردید برای محاسبه فاکتوریل نیز باید از یک حلقه بهره برد و چون این مقدار در هر جمله باید محاسبه شود پس باید این حلقه در داخل حلقه اول و با هر بار اجرای آن اجرا شود. به این‌گونه حلقه‌ها، حلقه‌های متداخل یا تودرتو می‌گویند. با توجه به مطالب گفته شده، الگوریتم زیر را می‌توان ارائه کرد:

۱- شروع

$$2- \quad x \text{ و } n \text{ را دریافت کن}$$

$$3- \quad I \leftarrow 1 \text{ و } K \leftarrow 1 \text{ و } SUM \leftarrow 0$$

$$4- \quad j \leftarrow 1 \text{ و } Fact \leftarrow 1$$

$$5- \quad Fact \leftarrow Fact \times j$$

$$6- \quad j \leftarrow j + 1$$

$$7- \quad \text{اگر } I \leq j \text{ آن‌گاه برو به مرحله ۵}$$

$$8- \quad SUM \leftarrow SUM + K \times \frac{x^I}{Fact}$$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۲۲/۲۷	شماره شناسایی: ۳-۲۲/۲۷-۲	شماره شناسایی: ۳-۲۲/۲۷-۲۱

$$I \leftarrow I + 2 \quad 9-$$

$$K \leftarrow K \times -1 \quad 10-$$

$$I \leq 2n + 1 \text{ اگر } I \text{ آن گاه برو به مرحله } 4 \quad 11-$$

12- SUM را نمایش بده

13- پایان

همان‌طور که مشاهده می‌کنید در الگوریتم فوق دو حلقه وجود دارد: یک حلقه که در مراحل ۵ تا ۷ اجرا می‌شود و برای محاسبه مقدار فاکتوریل به کار می‌رود و یک حلقه که در مراحل ۴ الی ۱۱ اجرا می‌شود و برای محاسبه جمع جملات به کار می‌رود بنابراین با هر بار اجرای حلقه بزرگ‌تر، حلقه داخلی یک‌بار به‌طور کامل اجرا می‌شود؛ به علاوه برای آن که ضریب هر جمله به‌طور یک در میان مثبت و منفی شود از متغیر K استفاده شده و مقدار اولیه آن ۱ قرار داده می‌شود. این متغیر در مرحله ۱۰، هر بار در مقدار ۱- ضرب می‌شود و سبب ایجاد ضریب ۱ و ۱- به‌طور متناوب خواهد شد. شرط خاتمه حلقه اصلی (بزرگ‌تر) نیز با توجه به مقدار n یا همان تعداد جملات محاسبه می‌شود.

تمرین: الگوریتمی بنویسید که عدد طبیعی و دلخواه N را دریافت کرده و معین کند عدد تام است یا خیر. عدد تام عددی است که مجموع مقسوم‌علیه‌های آن (به‌جز خود عدد) با خود عدد برابر باشد.

۳-۱ آرایه

الگوریتم‌هایی که تاکنون طراحی کرده‌اید داده‌ها را دریافت کرده و سپس بلافاصله مورد پردازش قرار می‌داد اما گاهی اوقات لازم است تا پردازش پس از ورود تمام یا بخشی از اطلاعات انجام شود یا عملیاتی چون مرتب‌سازی یا جستجوی اطلاعات در داده‌های وارد شده انجام گیرد. در مواردی هم حجم داده‌های ورودی زیاد است و برای ذخیره‌سازی آن‌ها باید از متغیرهای متعددی با نام‌های متفاوت استفاده کرد و سپس از آن‌ها در پردازش بهره برد که کاری دشوار و در اکثر موارد غیرممکن خواهد بود؛ به عنوان مثال فرض کنید می‌خواهید الگوریتمی بنویسید که داده‌های مربوط به مشخصات و نمرات ۱۰۰ دانش‌آموز یک مدرسه را ذخیره و پردازش نماید، تعریف متغیرهای موردنیاز به این تعداد عملاً کار بسیار دشواری است.

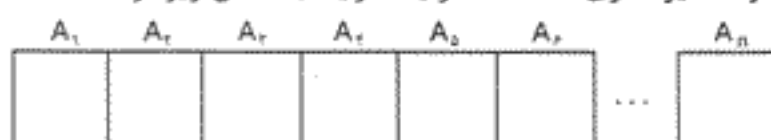
در چنین مواردی می‌توانید از یک مفهوم مهم که در ریاضیات با آن به شکل‌های مختلفی آشنا شده‌اید، استفاده کنید: آرایه، انجام این‌گونه عملیات را بسیار آسان و سریع می‌کند.

آرایه مکان‌هایی از حافظه است که با یک نام شناسایی شده و دارای اندیس‌هایی برای اشاره به خانه‌های این آرایه است. در نتیجه آرایه را می‌توان به صورت یک متغیر با مکان‌های متعددی که

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۲۲/۲۷	شماره شناسایی: ۳-۲۲/۲۷-۲	شماره شناسایی: ۳-۲۲/۲۷-۲۱

داده‌های هم‌نوع را در خود نگه می‌دارند، تعریف کرد. آدرس خانه‌های آرایه متوالی بوده و در حافظه به صورت پشت‌سرهم قرار می‌گیرند.

برای درک بهتر از نحوه ذخیره‌سازی اطلاعات در یک آرایه، به شکل زیر توجه کنید:



همان‌طور که مشاهده می‌کنید در شکل فوق یک آرایه با نام A و با تعداد n عضو ترسیم شده است و هر یک از فضاها به یکی از اعضای آرایه اختصاص دارد ولی تمام آن‌ها دارای نام A هستند که با استفاده از اندیس‌های ۱ تا n از یکدیگر متمایز می‌شوند. n یا تعداد اعضا به عهده شماست و می‌تواند یک مقدار ثابت یا متغیر باشد.

مثال ۱۸: الگوریتمی بنویسید که ۲۰ عدد دلخواه را دریافت کرده و در یک آرایه ذخیره نماید سپس آن‌ها را از آخر به اول نمایش دهد.

برای تعریف یک آرایه می‌توانید از دستورالعمل‌هایی مانند "متغیرهای A_1 و A_2 و A_3 و و A_{20} " یا "متغیرهای A_1 و A_2 و و A_{20} را ایجاد کن" استفاده و پس از تعریف آرایه باید به اعضای آن دسترسی پیدا کنید؛ برای این کار می‌توان هر یک از اعضا را به صورت انفرادی و با ذکر اندیس آن مورد دسترسی قرار داد یا با استفاده از یک حلقه و به کارگیری یک متغیر به عنوان اندیس و تغییر مقدار این متغیر، به هر یک از اعضا دسترسی پیدا کرد. با توجه به مطالب گفته شده می‌توان الگوریتم فوق را به صورت زیر طراحی کرد:

۱- شروع

۲- متغیرهای A_1 و A_2 و و A_{20} را در نظر بگیر

۳- $i \leftarrow 1$

۴- $A(i)$ را دریافت کن

۵- $i \leftarrow i + 1$

۶- اگر $i \leq 20$ آن‌گاه برو به مرحله ۴

۷- $i \leftarrow 20$

۸- $A(i)$ را نمایش بده

۹- $i \leftarrow i - 1$

۱۰- اگر $i \geq 1$ آن‌گاه برو به مرحله ۸

۱۱- پایان

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

همان‌طور که در این الگوریتم مشاهده می‌کنید ابتدا در مرحله ۲ آرایه A با ۲۰ عضو تعریف شده است سپس در مراحل ۴ تا ۶ با استفاده از یک حلقه اعضای آرایه یکی یکی پر می‌شود. این عمل با استفاده از متغیر i به عنوان اندیس انجام می‌شود. متغیر i ابتدا با مقدار ۱ (اندیس اولین عضو در آرایه A) شروع و سپس در داخل حلقه و در هر مرحله یک واحد به آن اضافه می‌شود تا در مرحله بعد عضو دوم و به همین صورت تا عضو بیستم را از کاربر دریافت کند سپس با استفاده از یک حلقه معکوس، اعضای آرایه از آخر به اول نمایش داده می‌شوند. برای این کار مقدار ۲۰ در متغیر i ذخیره می‌شود و پس از نمایش عضو بیستم یعنی A_{۲۰} یک واحد از متغیر i کاسته می‌شود، سپس با استفاده از یک دستور شرطی مراحل ۸ و ۹ تا رسیدن به اولین عضو آرایه (یعنی i=۱) ادامه یافته و پس از خاتمه اجرای حلقه، الگوریتم در مرحله ۱۱ پایان می‌یابد.

تمرین: الگوریتمی بنویسید که ۲۰۰ عدد دلخواه را به صورت یک آرایه دریافت کرده و بزرگ‌ترین و کوچک‌ترین عدد را در بین آن‌ها محاسبه کند و نمایش دهد.

مثال ۱۹: الگوریتمی بنویسید که دو آرایه یک بعدی با تعداد اعضای دلخواه را دریافت کرده و حاصل جمع آن‌ها را در آرایه یک بعدی جداگانه‌ای ذخیره نموده و نمایش دهد. همان‌طور که می‌دانید اگر آرایه A و B را با n عضو در نظر بگیرید آرایه حاصل جمع C به صورتی که در ادامه می‌آید، محاسبه می‌شود.

$$\begin{array}{r} C_1 = A_1 + B_1 \\ C_2 = A_2 + B_2 \\ \vdots \\ C_n = A_n + B_n \end{array}$$

پس از آن که نحوه محاسبه جمع دو آرایه یک بعدی را بررسی کردید، می‌توانید با استفاده از حلقه‌ها اعضای آرایه را دریافت و حاصل جمع را در آرایه دیگری ذخیره کنید.

نکته: دو آرایه را وقتی می‌توان جمع کرد که تعداد عناصر آن‌ها برابر باشد.



بنابراین الگوریتم مورد نظر به صورت زیر خواهد بود:

۱- شروع

۲- n را دریافت کن

۳- A_۱ و A_۲ و ... و A_n را در نظر بگیر

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

۴- B_n و ... و B_7 و B_1 را در نظر بگیر

۵- C_n و ... و C_7 و C_1 را در نظر بگیر

۶- $i \leftarrow 1$

۷- $A(i)$ را دریافت کن

۸- $i \leftarrow i + 1$

۹- اگر $i \leq n$ آن‌گاه برو به مرحله ۷

۱۰- $i \leftarrow 1$

۱۱- $B(i)$ را دریافت کن

۱۲- $i \leftarrow i + 1$

۱۳- اگر $i \leq n$ آن‌گاه برو به مرحله ۱۱

۱۴- $i \leftarrow 1$

۱۵- $C(i) \leftarrow A(i) + B(i)$

۱۶- $i \leftarrow i + 1$

۱۷- اگر $i \leq n$ آن‌گاه برو به مرحله ۱۵

۱۸- $i \leftarrow 1$

۱۹- $C(i)$ را نمایش بده

۲۰- $i \leftarrow i + 1$

۲۱- اگر $i \leq n$ آن‌گاه برو به مرحله ۱۹

۲۲- پایان

تمرین: الگوریتمی بنویسید که دو آرایه یک بعدی با تعدادی اعضای دلخواه را دریافت کرده، حاصل تفریق آن‌ها را محاسبه کند و در آرایه جداگانه‌ای ذخیره نماید؛ سپس آرایه را به همراه آرایه حاصل تفریق نمایش دهید.

مثال ۲۰: الگوریتمی بنویسید که با استفاده از مفهوم آرایه، ساعات کار کارمندان یک شرکت را در یک هفته دریافت کرده و سپس میزان حقوق هر یک از آن‌ها را براساس ۱۰۰۰۰ ریال برای هر ساعت محاسبه نموده و نمایش دهد (با فرض این که تعداد کارمندان ۵۰ نفر باشد و برای متمایز کردن کارمندان نیز یک شماره کارمندی در نظر گرفته شود).

برای حل این مسئله ملاحظه می‌کنید که در هر هفته برای هر کارمند ۶ روز کاری وجود دارد؛ برای طراحی این الگوریتم با استفاده از مفهوم آرایه یک بعدی لازم است تا برای هر کارمند یک آرایه با تعداد ۷ عضو تعریف کنیم تا یکی از اعضا برای ذخیره کردن شماره کارمندی هر یک از آن‌ها و ۶

واحد کار، حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

عضو دیگر برای ذخیره‌سازی ساعات کار آن‌ها در ۶ روز کاری هفته مورد استفاده قرار گیرند؛ به این ترتیب باید ۵۰ آرایه هر یک با ۷ عضو یا ۷ آرایه با ۵۰ عضو تعریف کنید.

اما راه حل بهتری نیز برای این گونه حالت‌ها وجود دارد و آن استفاده از آرایه‌ها یا لیست‌های دوبعدی است. در واقع یک آرایه دوبعدی از مجموعه چند سطر و ستون تشکیل می‌شود و هر سطر یک آرایه ساده (یک بعدی) است بنابراین اگر A یک آرایه با m سطر و n ستون باشد، هر عضو در آرایه با دو اندیس از دیگری متمایز می‌شود؛ اندیس اول شماره سطر و اندیس دوم شماره ستون را مشخص می‌کنند. با توجه به مطالبی که گفته شد یک آرایه دوبعدی را می‌توانید به صورت زیر در نظر بگیرید:

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

بنابراین برای حل این مثال می‌توانید از یک آرایه دوبعدی با ۵۰ سطر و ۷ ستون استفاده کنید، هر سطر را به اطلاعات یک کارمند اختصاص دهید و در هر سطر، اولین ستون را به شماره کارمندی و ۶ ستون بعدی را به ساعات کاری وی در هر هفته مختص کنید.

برای دسترسی به اعضای یک آرایه دوبعدی باید از دو حلقه در داخل یکدیگر استفاده کرد. حلقه اول کنترل سطرها و حلقه دوم کنترل ستون‌ها را به عهده می‌گیرند. با توجه به توضیحاتی که آرایه شد الگوریتم مورد نظر را می‌توان به این صورت نوشت:

۱- شروع

۲- آرایه $M_{50 \times 7}$ را در نظر بگیر

۳- $i \leftarrow 1$

۴- $j \leftarrow 1$

۵- $M_{(i,j)}$ را دریافت کن

۶- $j \leftarrow j + 1$

۷- اگر $j \leq 7$ آن‌گاه برو به مرحله ۵

۸- $i \leftarrow i + 1$

۹- اگر $i \leq 50$ آن‌گاه برو به مرحله ۴

۱۰- $i \leftarrow 1$

۱۱- $j \leftarrow 2$ و $SUM \leftarrow 0$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۳	شماره شناسایی: ۳-۴۲/۲۷-۲۱

$$SUM \leftarrow SUM + M_{(i,j)} - 12$$

$$j \leftarrow j + 1 - 13$$

$$14- \text{اگر } j \leq n \text{ آن‌گاه برو به مرحله } 12$$

$$15- M_{(i,n)} \text{ و } SUM \times 10000 \text{ را نمایش بده}$$

$$i \leftarrow i + 1 - 16$$

$$17- \text{اگر } i \leq 50 \text{ آن‌گاه برو به مرحله } 11$$

$$18- \text{پایان}$$

همان‌طور که مشاهده می‌کنید پس از تعریف آرایه ابتدا با استفاده از دو حلقه تودرتو در مراحل ۴ تا ۹، اعضای آرایه M دریافت می‌شوند. متغیر i برای ستون و j برای سطر مورد استفاده قرار می‌گیرند. پس از دریافت آرایه در مراحل ۱۱ تا ۱۷ با استفاده از دو حلقه تودرتوی دیگر مجموع ساعات کار هر کارمند محاسبه شده و در متغیر SUM ذخیره می‌شود، سپس در هر مرتبه اجرای حلقه داخلی خاتمه می‌یابد. در مرحله ۱۵ شماره کارمندی هر نفر $M_{(i,n)}$ به همراه مبلغ کارکرد هر کارمند در هفته محاسبه شده و نمایش داده می‌شود. در مرحله بعد (مرحله ۱۶) با اضافه کردن یک واحد به متغیر i و اجرای دستورات از مرحله ۱۱ به بعد، اطلاعات مربوط به نفر بعدی مورد پردازش قرار می‌گیرد تا اطلاعات تمام ۵۰ کارمند مورد محاسبه قرار گیرد.

تمرین: الگوریتم فوق را به گونه‌ای تنظیم کنید که توانایی محاسبه حقوق تعداد دلخواهی از کارمندان را برای یک ماه داشته باشد.

مثال ۲۱: الگوریتمی بنویسید که یک آرایه دوبعدی $(m \times n)$ را دریافت کرده و مجموع سطرها و ستون‌های آن را محاسبه کرده و نمایش دهد.

$$1- \text{شروع}$$

$$2- m \text{ و } n \text{ را دریافت کن}$$

$$3- \text{آرایه } A_{m \times n} \text{ را در نظر بگیر}$$

$$4- i \leftarrow 1$$

$$5- j \leftarrow 1$$

$$6- A_{(i,j)} \text{ را دریافت کن}$$

$$7- j \leftarrow j + 1$$

$$8- \text{اگر } j \leq n \text{ آن‌گاه برو به مرحله } 6$$

$$9- i \leftarrow i + 1$$

$$10- \text{اگر } i \leq m \text{ آن‌گاه برو به مرحله } 5$$

استاندارد مهارت: رایانه کار درجه ۱	بیانه مهارتی الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

۱۱- $\text{Sum} \leftarrow 0$

۱۲- $i \leftarrow 1$

۱۳- $j \leftarrow 1$

۱۴- $\text{Sum} \leftarrow \text{Sum} + A_{(i,j)}$

۱۵- $j \leftarrow j + 1$

۱۶- اگر $j \leq n$ آن‌گاه برو به مرحله ۱۴

۱۷- $i \leftarrow i + 1$

۱۸- اگر $i \leq m$ آن‌گاه برو به مرحله ۱۳

۱۹- Sum را نمایش بده

۲۰- پایان

همان‌طور که در الگوریتم فوق مشاهده می‌کنید، پس از دریافت تعداد سطرها (m) و تعداد ستون‌ها (n)، اعضای آرایه با استفاده از دو حلقه تودرتو دریافت شده و سپس مجدداً با به‌کارگیری دو حلقه تودرتوی دیگر مجموع اعضای آن در متغیر Sum ذخیره می‌شود و در مرحله ۱۹ این مقدار نمایش می‌یابد.

تمرین: الگوریتمی بنویسید که یک آرایه دو بعدی $m \times m$ را دریافت کرده و مجموع اعضای روی قطر اصلی آن را محاسبه کرده و نمایش دهد.

نکته: قطر اصلی، مجموعه‌ای از اعضای آرایه است که مقدار اندیس‌های سطر و ستون آن‌ها با هم برابر است، مثلاً در آرایه دو بعدی $A_{(i,j)}$ اعضای $A_{(1,1)}$ ، $A_{(2,2)}$ و $A_{(3,3)}$ قطر اصلی را تشکیل می‌دهند.

۴-۱ زیر الگوریتم

در حل مسایل واقعی الگوریتم، معمولاً با تعدادی دستورالعمل که بارها در الگوریتم استفاده شده‌اند، مواجه می‌شوید که طراحی و نوشتن چنین الگوریتمی کار بسیار دشوار و زمان‌بری است. در چنین شرایطی بهتر است تا این بخش‌ها به صورت جداگانه طراحی و نوشته شوند و در هر مرحله از الگوریتم مورد نظر مورد استفاده قرار گیرند. برای انجام این کار نیاز به مفهوم دیگری به نام زیر الگوریتم خواهید داشت. زیر الگوریتم در واقع یک الگوریتم است که به حل مسأله در الگوریتم اصلی کمک می‌کند. استفاده از زیر الگوریتم مزایای زیر را دارد:

۱- امکان تعمیم و ارتقای آسان‌تر الگوریتم در آینده

۲- جلوگیری از تکرار دستورالعمل‌ها

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

۳- خطایابی و خطازدایی آسان‌تر و سریع‌تر

۴- استفاده از زیرالگوریتم در الگوریتم‌های مشابه

برای درک بهتر از زیرالگوریتم به مثالی در این زمینه توجه کنید.

مثال ۲۲: الگوریتم محاسبه حاصل‌جمع را که در مباحث قبلی حل کرده‌اید با استفاده از زیر الگوریتم مجدداً طراحی کنید:

$$\text{Sum} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^{(2n+1)}}{(2n+1)!}$$

در الگوریتم ارائه شده، بخش محاسبه فاکتوریل را از الگوریتم اصلی جدا کرده و به صورت زیرالگوریتم تعریف کنید سپس زیرالگوریتم را در هر مرحله در الگوریتم اصلی فراخوانی کنید، بنابراین زیرالگوریتم و الگوریتم مربوطه به صورت زیر خواهد بود:

زیرالگوریتم Factorial(Fact,I)

۱- شروع

۲- $I \leftarrow J$

۳- $\text{Fact} \leftarrow \text{Fact} \times J$

۴- $J \leftarrow J - 1$

۵- اگر $1 \leq J$ آن‌گاه برو به مرحله ۳

۶- بازگشت

الگوریتم اصلی

۱- شروع

۲- x و n را دریافت کن

۳- $I \leftarrow 1$ و $K \leftarrow 1$ و $\text{SUM} \leftarrow 0$

۴- $\text{Fact} \leftarrow 1$

۵- زیرالگوریتم Factorial (Fact, I) را اجرا کن

۶- $\text{SUM} \leftarrow \text{SUM} + K \times \frac{x^I}{\text{Fact}}$

۷- $I \leftarrow I + 2$

۸- $K \leftarrow K \times -1$

۹- اگر $I \leq 2n + 1$ آن‌گاه برو به مرحله ۴

۱۰- SUM را نمایش بده

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

۱۱- پایان

اولین قدم برای طراحی زیرالگوریتم، انتخاب یک نام برای آن است تا به وسیله آن بتوان در هرجایی که لازم باشد آن را فراخوانی کرد، به علاوه با توجه به راه حل مسأله و وظایفی که به عهده زیرالگوریتم گذاشته می‌شود، ممکن است مقادیر یا متغیرهایی به عنوان داده‌های ورودی یا اطلاعات بازگشتی به زیرالگوریتم وارد یا از آن خارج شوند؛ به این متغیرها آرگومان می‌گویند. تعداد آرگومان‌ها توسط طراح الگوریتم تعیین می‌شود و نام آن‌ها در داخل یک جفت پرانتز در جلوی نام زیرالگوریتم نوشته شده و با کاراکتر کاما از یکدیگر متمایز می‌شوند. در زیرالگوریتم Factorial دو آرگومان وجود دارد: آرگومان Fact که مقدار فاکتوریل محاسبه شده را به محل فراخوانی زیرالگوریتم باز می‌گرداند و آرگومان I که مقداری را که فاکتوریل آن به وسیله زیرالگوریتم محاسبه می‌شود به زیرالگوریتم انتقال می‌دهد. پس از تعیین نام و آرگومان‌های زیرالگوریتم، دستورالعمل شروع و سایر دستورالعمل‌های مورد نیاز قرار می‌گیرند. برای تعیین محل پایان و خاتمه اجرای زیرالگوریتم از دستورالعمل بازگشت استفاده می‌شود تا اجرای الگوریتم به محل فراخوانی زیرالگوریتم یعنی الگوریتم اصلی منتقل شود.

پس از خاتمه طراحی زیرالگوریتم، باید آن را در الگوریتم اصلی فراخوانی کرد؛ به این منظور با استفاده از نام زیرالگوریتم می‌توان زیرالگوریتم را در الگوریتم اجرا یا به عبارت دیگر فراخوانی کرد. اکنون اگر الگوریتم را اجرا کنید با رسیدن به مرحله ۵ و اجرای زیرالگوریتم Factorial، اجرای الگوریتم به زیرالگوریتم Factorial انتقال می‌یابد و متغیرهای I و Fact در اختیار آن قرار می‌گیرند، سپس با اجرای مراحل ۱ تا ۵ در زیرالگوریتم Factorial مقدار فاکتوریل I محاسبه شده و با رسیدن به مرحله ۶ و دستورالعمل «بازگشت»، مجدداً به الگوریتم اصلی بازگشته و سپس مرحله ۶ به بعد اجرا می‌شود. با تکرار این عملیات مقدار سینوس زاویه، محاسبه و با خاتمه حلقه موجود در الگوریتم اصلی مقدار محاسبه شده به نمایش در می‌آید و در مرحله ۱۱ الگوریتم خاتمه می‌یابد.

تمرین: الگوریتمی بنویسید که با استفاده از یک زیرالگوریتم، مجموعه زیر را محاسبه کرده و نمایش دهد (با فرض آن که $n > 0$).

$$\text{Sum} = \frac{2}{x^2 + 2!} + \frac{4}{x^4 + 4!} + \dots + \frac{2n}{x^{2n} + 2n!}$$

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۳۱

مخصوص رشته برنامه‌نویسی

۵-۱ روش‌های مرتب‌سازی داده‌ها و اطلاعات

یکی از مسائل دیگری که در پردازش داده‌ها و اطلاعات مورد توجه قرار می‌گیرد، مرتب کردن داده‌ها و اطلاعات است؛ البته روش‌های متعددی برای انجام این کار وجود دارد که استفاده از هریک از آن‌ها معایب و مزایایی را به دنبال دارد. یکی از نکات مهمی که در مرتب‌سازی باید مورد توجه قرار گیرد استفاده از آرایه‌ها برای مرتب کردن داده و اطلاعات است. بدون استفاده از مفهوم آرایه عملاً امکان استفاده از الگوریتم‌هایی که آرایه می‌شود، وجود نخواهد داشت. در دنیای الگوریتم روش‌های مختلفی برای انجام عملیات مرتب‌سازی وجود دارد که به عنوان نمونه می‌توان به روش‌های مرتب‌سازی حبابی و سریع اشاره کرد. در هر یک از این روش‌ها از تکنیک متفاوتی استفاده می‌شود. در این جا به توضیح روش حبابی که ساده‌تر است، می‌پردازیم:

۵-۱-۱ روش مرتب‌سازی حبابی (Bubble Sort)

این روش، ساده‌ترین روش مرتب‌سازی است و هر عضو از آرایه با اعضای بعدی آن در آرایه مقایسه می‌شود و در صورت لزوم عمل تعویض آن‌ها صورت می‌گیرد؛ به عنوان مثال اعداد زیر را در نظر بگیرید:

۵ ۲ ۴ ۱۰ ۷

برای مرتب‌سازی اعداد فوق به روش حبابی (صعودی) ابتدا باید عدد ۵ با ۲ مقایسه شود و چون عدد ۵ از ۲ بزرگ‌تر است، بنابراین موقعیت عدد ۵ با ۲ به این صورت عوض می‌شود:

۲ ۵ ۴ ۱۰ ۷

در مرحله بعد عدد ۵ با ۴ مقایسه می‌شود و چون عدد ۵ بزرگ‌تر است، بنابراین موقعیت عدد ۵ با ۴ عوض می‌شود.

۲ ۴ ۵ ۱۰ ۷

به همین شکل عدد ۵ با ۱۰ نیز مقایسه می‌شود و چون کوچک‌تر از آن است، بنابراین تعویض اعداد انجام نمی‌گیرد. در مرحله آخر عدد ۱۰ با ۷ مقایسه شده و چون عدد ۱۰ از ۷ بزرگ‌تر است، عمل جابه‌جایی انجام می‌شود.

۲ ۴ ۵ ۷ ۱۰

در دور دوم عدد ۲ با ۴ و سپس ۴ با ۵ مقایسه می‌شود و چون عدد ۲ از ۴ و ۴ از ۵ کوچک‌تر است، عمل جابه‌جایی انجام نمی‌شود و به همین شکل عدد ۵ با ۷ و ۷ با ۱۰ نیز مقایسه می‌شود. با انجام مقایسه‌ها در دور سوم و چهارم نیز هر عدد با عدد بعدی مقایسه و در صورت لزوم جابه‌جا

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

می‌شوند؛ در این صورت برای مرتب کردن ۵ عدد، ۴ دور مقایسه انجام می‌شود که در هر بار تمام اعداد کنار هم نیز با هم مقایسه می‌شوند یعنی در هر بار ۴ مرحله مقایسه انجام می‌گیرد. بنابراین با استفاده از دو حلقه تودرتو می‌توان الگوریتم مرتب‌سازی را طراحی کرد.

همان‌طور که مشاهده کردید، ممکن است داده‌ها بدون آن که تمام مقایسه‌ها انجام گیرد با اجرای چند مرحله مرتب شوند اما عملیات مقایسه تا پایان انجام می‌پذیرند. این مسأله یکی از معایب روش مرتب‌سازی حبابی است و عیب دیگر آن است که به دلیل انجام تعداد زیاد مقایسه‌ها برای مرتب کردن داده‌ها با تعداد زیاد، زمان زیادی مورد نیاز است.

مثال ۲۳: الگوریتمی بنویسید که تعداد دلخواهی عدد را دریافت کرده و با استفاده از روش مرتب‌سازی حبابی آن را به صورت صعودی مرتب کرده، نمایش دهد.

۱- شروع

۲- n را دریافت کن

۳- $M_1, M_2, \dots, M_n$ و M_i را در نظر بگیر

۴- $i \leftarrow 1$

۵- $M(i)$ را دریافت کن

۶- $i \leftarrow i + 1$

۷- اگر $i \leq n$ آن‌گاه برو به مرحله ۵

۸- $j \leftarrow 1$

۹- $k \leftarrow 1$

۱۰- اگر $M_{(k)} \leq M_{(k+1)}$ آن‌گاه برو به مرحله ۱۴

۱۱- $temp \leftarrow M_{(k)}$

۱۲- $M_{(k)} \leftarrow M_{(k+1)}$

۱۳- $M_{(k+1)} \leftarrow temp$

۱۴- $k \leftarrow k + 1$

۱۵- اگر $k \leq n - 1$ آن‌گاه برو به مرحله ۱۰

۱۶- $j \leftarrow j + 1$

۱۷- اگر $j \leq n - 1$ آن‌گاه برو به مرحله ۹

۱۸- $i \leftarrow 1$

۱۹- $M(i)$ را نمایش بده

۲۰- $i \leftarrow i + 1$

۲۱- اگر $i \leq n$ آن‌گاه برو به مرحله ۱۹

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۲-۴۲/۲۷	شماره شناسایی: ۲-۴۲/۲۷-۲	شماره شناسایی: ۲-۴۲/۲۷-۲۱

۲۲- پایان

همان‌طور که ملاحظه می‌کنید در این الگوریتم پس از دریافت آرایه M با n عضو عملیات مرتب‌سازی از مرحله ۸ آغاز شده و به این منظور از دو متغیر z و k استفاده می‌شود. در مرحله ۱۰ با استفاده از یک دستورالعمل شرطی اولین عضو آرایه با عضو دوم مقایسه می‌شود. اگر عضو اول کوچک‌تر یا مساوی عضو دوم باشد، اجرای الگوریتم به مرحله ۱۴ منتقل می‌شود و در غیراین صورت با استفاده از یک متغیر واسطه $temp$ عضو اول با دوم جابه‌جا می‌شود. در هر دو حالت با رسیدن به مرحله ۱۴ یک واحد به متغیر k اضافه می‌شود تا در مرحله بعد عضو دوم با سوم مقایسه شود. برای ایجاد حلقه اول در مرحله ۱۵ از دستورالعمل شرطی استفاده می‌شود و تا زمانی که دو عضو آخر با هم مقایسه نشوند ($k \leq n-1$)، اجرای الگوریتم به مرحله ۱۰ منتقل می‌شود. پس از انجام دور اول از مقایسه‌ها مرحله ۱۶ اجرا می‌شود و با استفاده از یک دستورالعمل شرطی، دور دوم از مقایسه‌ها از مرحله ۹ مجدداً اجرا خواهد شد؛ به این صورت با خاتمه اجرای حلقه‌ها آرایه به صورت صعودی مرتب خواهد شد.

تمرین: الگوریتمی بنویسید که تعداد دلخواهی عدد را دریافت کند سپس با استفاده از روش مرتب‌سازی حبابی آن‌را به صورت نزولی مرتب کرده و نمایش دهد.

۶-۱ روش‌های جستجوی داده‌ها و اطلاعات

یکی از مسایل دیگری که در طراحی الگوریتم‌ها مورد توجه قرار می‌گیرد جستجو و پیدا کردن داده یا داده‌هایی در میان مجموعه‌ای از اطلاعات موجود است. به این منظور مانند نحوه مرتب‌سازی داده و اطلاعات، روش‌های مختلفی نیز برای جستجوی داده‌های موردنظر وجود دارد که از مهم‌ترین آن‌ها می‌توان به روش جستجوی خطی (Linear) و روش جستجوی دودویی (Binary) اشاره کرد.

۶-۱-۱ روش جستجوی خطی

این روش یکی از ساده‌ترین روش‌های جستجو است؛ در این روش برای پیدا کردن یک داده، آن را با تک‌تک اعضای آرایه مورد مقایسه قرار داده و در صورت پیدا شدن آن جستجو خاتمه می‌یابد و اگر داده موردنظر در آرایه پیدا نشود، نتیجه جستجو منفی خواهد بود. به این منظور می‌توانید از یک حلقه و دستورالعمل شرطی استفاده کنید.

مثال ۲۴: الگوریتمی بنویسید که یک آرایه عددی n عضوی و یک عدد دلخواه را دریافت کرده و عدد مربوطه را در آرایه جستجو کند.

استاندارد مهارت، رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

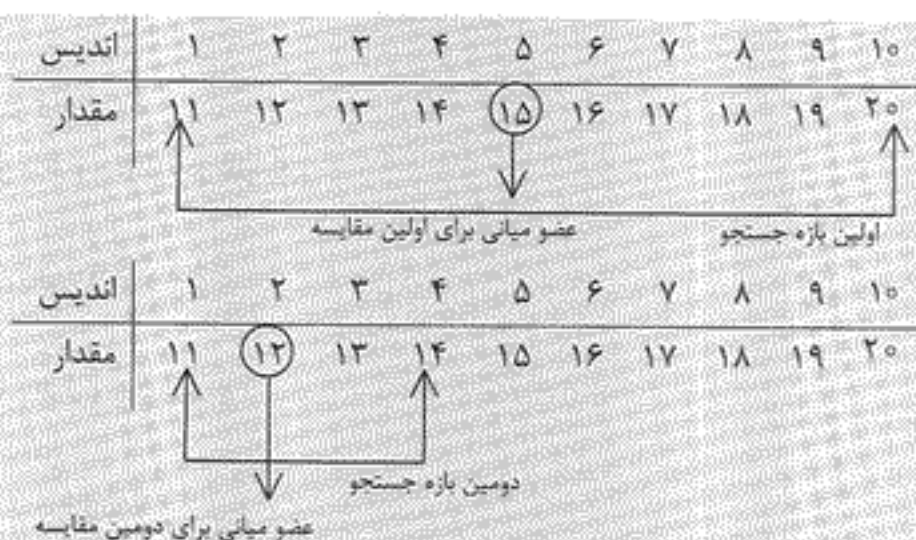
- ۱- شروع
- ۲- n را دریافت کن
- ۳- M_1 و M_2 و ... و M_n را در نظر بگیر
- ۴- $i \leftarrow 1$
- ۵- $M_{(i)}$ را دریافت کن
- ۶- $i \leftarrow i + 1$
- ۷- اگر $i \leq n$ آن‌گاه برو به مرحله ۵
- ۸- S را دریافت کن
- ۹- $i \leftarrow 1$ و $Found \leftarrow 0$
- ۱۰- اگر $S = M_{(i)}$ آن‌گاه $Found \leftarrow i$ و $i \leftarrow n$
- ۱۱- $i \leftarrow i + 1$
- ۱۲- اگر $i \leq n$ آن‌گاه برو به مرحله ۱۰
- ۱۳- اگر $Found = 0$ آن‌گاه "عدد در آرایه وجود ندارد!" را نمایش بده در غیر این صورت $Found$ و M_{Found} را نمایش بده
- ۱۴- پایان

تمرین: الگوریتمی بنویسید که تعداد دلخواهی عدد را به صورت آرایه دریافت کند و سپس یک عدد دلخواه را در آن جستجو کرده و تعداد دفعات تکرار آن را در آرایه معین کند.

۲-۶-۱ روش جستجوی باینری (دودویی)

روش جستجوی خطی در آرایه‌های بزرگ بسیار کند و وقت‌گیر است، با استفاده از روش جستجوی باینری یا دودویی می‌توانید تعداد مقایسه‌ها و در نتیجه زمان لازم برای پیدا کردن داده‌ها و اطلاعات موردنظر را به میزان قابل توجهی کاهش دهید. در این روش ابتدا آرایه را مرتب کرده و سپس عضو میانی آرایه با داده موردنظر برای جستجو، مقایسه شود. اگر داده موردنظر کوچک‌تر از عضو میانی باشد، جستجو در قسمت پایینی عضو میانی صورت می‌گیرد و اگر داده مورد جستجو بزرگ‌تر از عضو میانی باشد، جستجو در قسمت بالایی عضو میانی انجام می‌شود. این عملیات برای نیمه پایینی یا بالایی در آرایه نیز انجام می‌شود و تا زمان پیدا شدن داده مورد جستجو با نصف شدن تعداد اعضای باقیمانده ادامه می‌یابد. به عنوان مثال نمودار جستجوی عدد ۱۲، در آرایه‌ای با اعضای ۱۱ تا ۲۰ به صورت بعد خواهد بود:

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و هنجار	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱



مثال ۲۵: الگوریتم مثال قبل را با استفاده از روش دودویی بنویسید.

- ۱- شروع
- ۲- n را دریافت کن
- ۳- M_n و M_1 و M_2 را در نظر بگیر
- ۴- $i \leftarrow 1$
- ۵- $M_{(i)}$ را دریافت کن
- ۶- $i \leftarrow i + 1$
- ۷- اگر $i \leq n$ آن گاه برو به مرحله ۵
- ۸- S را دریافت کن
- ۹- $Found \leftarrow 0$
- ۱۰- $low \leftarrow 1$ و $high \leftarrow n$
- ۱۱- $mid \leftarrow (low + high) \div 2$
- ۱۲- اگر $S = M_{(mid)}$ آن گاه $Found \leftarrow 1$ و برو به مرحله ۱۶
- ۱۳- اگر $S > M_{(mid)}$ آن گاه $low \leftarrow mid + 1$
- ۱۴- اگر $S < M_{(mid)}$ آن گاه $high \leftarrow mid - 1$
- ۱۵- اگر $low \leq high$ آن گاه برو به مرحله ۱۱
- ۱۶- اگر $Found = 0$ آن گاه "عدد در آرایه وجود ندارد!" را نمایش بده در غیر این صورت mid و $M_{(mid)}$ را نمایش بده
- ۱۷- پایان

تمرین: الگوریتم تمرین قبل را با استفاده از روش جستجوی باینری بنویسید.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۲۲/۲۷	شماره شناسایی: ۳-۲۲/۲۷-۲	شماره شناسایی: ۳-۲۲/۲۷-۳۱

۷-۱ پشته (Stack)

تاکنون از روش‌های مختلفی برای ذخیره‌سازی داده‌ها و اطلاعات استفاده کرده‌اید. علاوه بر روش‌های مذکور با استفاده از روش‌های دیگری نیز می‌توان داده‌ها و اطلاعات موردنظر خود را در حافظه ذخیره کرد؛ یکی از این روش‌ها استفاده از مفهوم پشته است.

پشته در واقع لیستی از داده‌هاست و هر داده به‌دنبال داده دیگر قرار گرفته یا به عبارت دیگر نوشته می‌شود. برای استفاده از آن‌ها باید به ترتیب، ابتدا آخرین داده خوانده شده یا به عبارت دیگر از روی داده‌های دیگر خارج شود، سپس داده ذخیره شده قبل از آن قابل استفاده شود. به عنوان مثال می‌توان پشته را مانند میله‌ای تصور کرد که دیسک‌های مختلفی به ترتیب روی آن قرار می‌گیرند. در واقع هر دیسک روی دیسک قبلی قرار داده می‌شود و اگر بخواهید دیسکی را از میله خارج کنید باید دیسک‌های روی آن را قبلاً خارج کرده باشید. برای درک بهتر از نحوه ذخیره‌سازی داده در پشته به شکل زیر توجه کنید:

ذخیره کردن اولین داده	داده شماره ۱			
ذخیره کردن دومین داده	داده شماره ۱	داده شماره ۲		
ذخیره کردن سومین داده	داده شماره ۱	داده شماره ۲	داده شماره ۳	

هم‌چنین خواندن داده‌ها از پشته به این صورت خواهد بود:

خواندن سومین داده	داده شماره ۱	داده شماره ۲	داده شماره ۳	
خواندن دومین داده	داده شماره ۱	داده شماره ۲		
خواندن اولین داده	داده شماره ۱			

مثال ۲۶: یک زیرالگوریتم بنویسید که با استفاده از مفهوم پشته مقدار فاکتوریل عدد طبیعی و دلخواه n را محاسبه کند.

برای حل این مسأله باید زیرالگوریتم به‌گونه‌ای طراحی شود که برای محاسبه فاکتوریل عدد n زیرالگوریتم خود را در داخل خود فراخوانی نماید. به این منظور زیرالگوریتم Factorial را بدین صورت بنویسید:

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

زیرالگوریتم (n, fact) Factorial

توضیح	
۱- شروع	
۲- اگر $n=0$ آن گاه $fact \leftarrow 1$ و بازگشت	بازگشت به محل فراخوانی زیرالگوریتم
۳- زیرالگوریتم (fact و $n-1$) Factorial را اجرا کن	
۴- $fact \leftarrow n \times fact$	
۵- بازگشت	بازگشت به محل فراخوانی زیرالگوریتم

به عنوان مثال اگر این زیرالگوریتم را به صورت (fact و ۵) Factorial اجرا کنید در این صورت پس از اجرای مرحله اول، دستورالعمل شرطی در مرحله ۲ اجرا می‌شود و چون $n=5$ و برابر یا صفر نیست، دستورالعمل موجود در بخش آن گاه اجرا نشده و مرحله ۳ انجام می‌شود. در این مرحله زیرالگوریتم Factorial مجدداً به صورت (fact و ۴) Factorial فراخوانی می‌شود و زیرالگوریتم از اولین دستور اجرا می‌شود. این عملیات آن قدر تکرار می‌شود تا $n=0$ شود، فراخوانی‌های انجام شده را می‌توان به صورت زیر نمایش داد. مقادیر مربوط به هر مرحله از فراخوانی در پشته ذخیره می‌شوند.

Factorial (۵ و fact)

Factorial (۴ و fact)

Factorial (۳ و fact)

Factorial (۲ و fact)

Factorial (۱ و fact)

Factorial (۰ و fact)

با انجام آخرین فراخوانی، نتیجه شرط $n=0$ در مرحله ۲ درست خواهد بود بنابراین مقدار ۱ در متغیر fact ذخیره و دستور بازگشت، اجرای الگوریتم را به مرحله فراخوانی قبل بازمی‌گرداند و با اجرای مرحله ۴، دستورالعمل $fact \leftarrow n \times fact$ اجرا می‌شود. با رسیدن به مرحله ۵ اجرای الگوریتم به محل فراخوانی قبل بازمی‌گردد تا مقدار ۵! محاسبه شود. نحوه اجرای این عملیات برعکس حالت فراخوانی و به صورت زیر خواهد بود:

$fact \leftarrow 5 \times 4 \times 3 \times 2 \times 1$

Factorial (۵ و fact)

$fact \leftarrow 4 \times 3 \times 2 \times 1$

Factorial (۴ و fact)

$fact \leftarrow 3 \times 2 \times 1$

Factorial (۳ و fact)

$fact \leftarrow 2 \times 1$

Factorial (۲ و fact)

$fact \leftarrow 1 \times 1$

Factorial (۱ و fact)

$fact \leftarrow 1$

Factorial (۰ و fact)

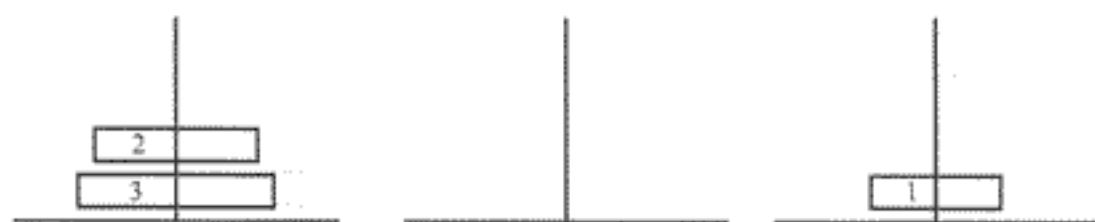
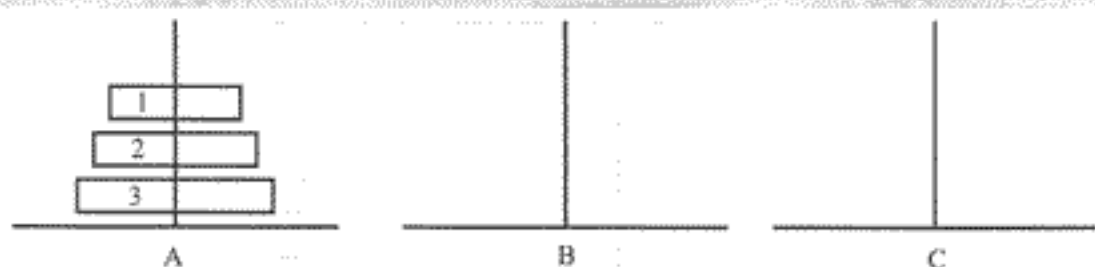
استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

همان‌طور که می‌بینید فراخوانی‌های انجام شده از پایین به بالا مقادیر محاسبه شده برای متغیر fact را باز می‌گرداند تا در پایان عدد ۱۲۰ به دست آمده و به محل فراخوانی اولیه یعنی (fact و ۵) Factorial بازگردانده شود. به این نوع از زیرالگوریتم‌ها، زیرالگوریتم‌های بازگشتی (Recursive) نیز می‌گویند.

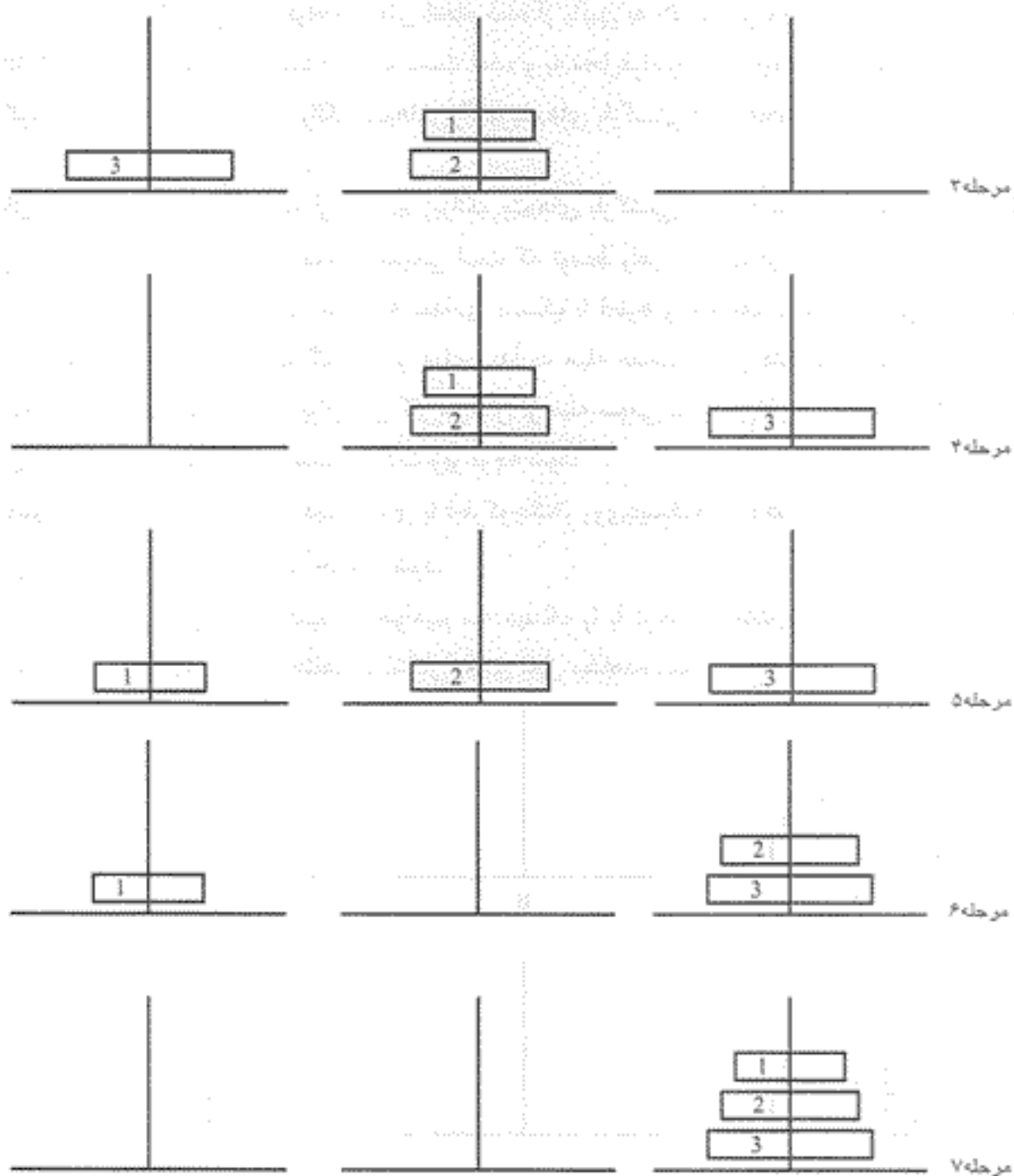
مثال ۲۷: می‌خواهیم با استفاده از مفاهیم زیرالگوریتم‌های بازگشتی و پشته یکی از مسایل قدیمی را حل کنیم. برج‌های هانوی یک معمای قدیمی است که توسط راهبان معابد در گذشته‌های دور مطرح شده است. مسأله به این صورت است که تعدادی دیسک با قطرهای مختلف به ترتیب روی یک میله قرار دارند. باید دیسک‌ها را به گونه‌ای از میله مبدأ به میله مقصد انتقال داد که ترتیب قرار گرفتن دیسک‌ها روی هم رعایت شود. برای این کار می‌توان از میله سومی به عنوان میله واسطه استفاده کرد. برای جابه‌جایی دیسک‌ها لازم است به نکات زیر توجه شود:

– همواره و در تمام مراحل باید دیسک‌های با قطر کوچک‌تر روی دیسک‌های با قطر بزرگ‌تر قرار بگیرند.
– در هر مرحله فقط یک دیسک جابه‌جا شود.

به عنوان مثال فرض کنید می‌خواهیم سه دیسک را با توجه به نکات ذکر شده از میله A به میله C منتقل کنیم، مراحل مختلف در اشکال بعدی قابل مشاهده است:



استاندارد مهارت: رایانه کار درجه ۱	بیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱



همان‌طور که ملاحظه می‌کنید عمل انتقال دیسک‌ها در هفت مرحله انجام می‌گیرد. با زیادتر شدن تعداد دیسک‌ها تعداد مراحل هم به شکل قابل توجهی افزایش خواهد یافت. برای طراحی زیرالگوریتمی که بتواند این مراحل را شبیه‌سازی کند باید از مفاهیم پشته و بازگشت پذیری استفاده کنیم. در غیر این صورت امکان آرایه یک زیرالگوریتم با روش‌های معمول وجود نخواهد داشت.

واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها	پسمانه مهنی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

اما برای رسیدن به راه حل مناسب توجه به مراحل که در انتقال سه دیسک وجود دارد، لازم است در واقع برای انتقال بزرگ‌ترین دیسک به میله مقصد باید تمام دیسک‌های کوچک‌تر در میله واسطه (B) قرار داده شوند و همین مسأله برای دیسک‌های بعدی نیز وجود دارد؛ بنابراین در زیرالگوریتمی که ارائه می‌شود، این نکته باید در نظر گرفته شود.

با توجه به مطالبی که گفته شد، زیرالگوریتم Tower به صورت زیر خواهد بود:

Tower (start, target, center, n)

زیرالگوریتم Tower (start, target, center, n)

۱- شروع

۲- اگر $n=1$ آن گاه

الف - یک دیسک را از میله start به میله target انتقال بده و بازگشت

در غیر این صورت

ب - زیرالگوریتم Tower (start, center, target, n-1) را اجرا کن

ج - یک دیسک از میله start به target انتقال بده

د - زیرالگوریتم Tower (center, target, start, n-1) را اجرا کن

۳- بازگشت

همان‌طور که ملاحظه می‌کنید، در زیرالگوریتم Tower از چهار آرگومان استفاده شده است. آرگومان‌های start, target و center به ترتیب اسامی میله‌های مبدأ، مقصد و واسطه و آرگومان n تعداد دیسک‌ها را دریافت می‌کند. برای درک بهتر عملکرد این زیرالگوریتم آن را برای حالت $n=3$ اجرا و مورد بررسی قرار می‌دهیم، بنابراین فراخوانی زیرالگوریتم به صورت زیر انجام می‌شود:

Tower(A,C,B,3)

پس از شروع الگوریتم و در مرحله دوم تعداد دیسک‌ها بررسی خواهند شد و چون نتیجه شرط نادرست است، دستورات بخش Else مورد استفاده قرار گرفته و اجرا می‌شوند، بنابراین فراخوانی زیر انجام می‌شود:

Tower(A,B,C,2)

سپس مجدداً زیرالگوریتم فراخوانی خواهد شد و در فراخوانی مجدد شرط $n=1$ نادرست بوده و فراخوانی دیگری اتفاق می‌افتد:

Tower(A,C,B,1)

با انجام این فراخوانی یک بار دیگر زیرالگوریتم اجرا می‌شود اما چون در این مرحله $n=1$ است، بنابراین یک دیسک (شماره ۱) از میله A به میله C منتقل خواهد شد، سپس عمل بازگشت انجام

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	بیمه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۳۱	شماره شناسایی: ۳-۴۲/۲۷-۳۰	شماره شناسایی: ۳-۴۲/۲۷

می‌گیرد و دستور بعد از آخرین فراخوانی یعنی دستور «ج» اجرا می‌شود و یک دیسک (شماره ۲) را از میله A به میله B انتقال می‌دهد. با اجرای دستور «د» یک فراخوانی دیگر به صورت زیر انجام می‌شود:

$Tower(C, B, A, 2-1)$

در نتیجه فراخوانی مجدد، شرط $n=1$ بررسی شده و چون نتیجه درست حاصل می‌شود، با اجرای دستور «الف» یک دیسک (شماره ۱) از میله C به میله B انتقال یافته و عمل بازگشت انجام می‌شود. با اجرای زیرالگوریتم تا این جا به مرحله ۳ در شکل‌های ارائه شده رسیده‌ایم. با انجام عمل بازگشت دستور «ج» اجرا شد و یک دیسک (شماره ۳) از میله A به میله C انتقال یافت سپس با اجرای دستور «د» یک فراخوانی دیگر به صورت زیر انجام می‌گیرد:

$Tower(B, C, A, 3-1)$

با فراخوانی مجدد زیرالگوریتم، بار دیگر دستورات موجود در بخش Else دستور شرطی موجود در مرحله ۲ زیرالگوریتم اجرا می‌شوند تا عملیات انتقال را کامل کنند. برای درک بهتر مطلب مراحل فراخوانی و انتقال دیسک‌ها را مشاهده کنید:

$Tower(A, C, B, 3)$

$Tower(A, B, C, 3-1)$

$Tower(A, C, B, 3-1)$

انتقال دیسک شماره ۱ از A به C و بازگشت

انتقال دیسک شماره ۲ از A به B و بازگشت

$Tower(C, B, A, 2-1)$

انتقال دیسک شماره ۱ از C به B و بازگشت

انتقال دیسک شماره ۳ از A به C و بازگشت

$Tower(B, C, A, 3-1)$

$Tower(B, A, C, 3-1)$

انتقال دیسک شماره ۲ از B به A و بازگشت

انتقال دیسک شماره ۱ از B به C و بازگشت

$Tower(A, C, B, 2-1)$

انتقال دیسک شماره ۲ از A به C و بازگشت

انتقال دیسک شماره ۱ از B به C و بازگشت

تمرین: الگوریتمی بنویسید که با استفاده از یک زیرالگوریتم بازگشتی، دنباله فیبوناچی را محاسبه کند. این دنباله به صورتی که در ادامه می‌آید، محاسبه می‌شود:

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

$$f(0) = 0$$

$$f(1) = 1$$

$$f(n) = f_{(n-1)} + f_{(n-2)} \quad n > 1$$

۸-۱ صف (Queue)

در زندگی روزمره با مفهوم صف آشنا هستید و بارها در صف‌های مختلف مانند صف نان، اتوبوس و مانند آن‌ها بوده‌اید. در تمام صف‌ها هر فرد جدیدی که به صف وارد می‌شود به انتهای صف اضافه شده و هر فردی که در صف قرار می‌گیرد تا رسیدن به ابتدای صف منتظر می‌ماند و وقتی به ابتدای صف می‌رسد، از صف خارج می‌شود.

یکی دیگر از روش‌های ذخیره‌سازی داده‌ها و اطلاعات، استفاده از مفهوم صف است. یک صف لیستی از داده‌هاست که اعضای آن به انتهای لیست اضافه شده و از ابتدای آن خارج می‌شوند، در واقع یک صف از داده‌ها مفهومی مشابه آرایه و پشته را دارد با این تفاوت که در آرایه می‌توان به هر یک از اعضا به طور مستقیم دسترسی داشت و در پشته، داده‌ها به انتهای لیست اضافه می‌شوند و یکی یکی از انتهای آن خارج می‌شوند. در صف فقط می‌توان به عضوی که در ابتدای صف قرار دارد، دسترسی پیدا کرد. البته باید توجه داشته باشید که برای مدیریت داده در یک صف باید از دو متغیر که یکی به عضو ابتدا و دیگری به عضو انتهایی در صف اشاره می‌کند، استفاده کرد. به عنوان مثال فرض کنید می‌خواهیم اسامی افرادی را در یک صف ذخیره و بازیابی کنیم، بنابراین قرار گرفتن و خارج کردن اسامی به این صورت خواهد بود:

Start = ۱
End = ۱

Ali	
-----	--

۱

Start = ۱
End = ۲

Ali	Reza	
-----	------	--

۱

۲

Start = ۱
End = ۲

Ali	Reza	Ahmad	
-----	------	-------	--

۱

۲

۳

Start = ۲
End = ۳

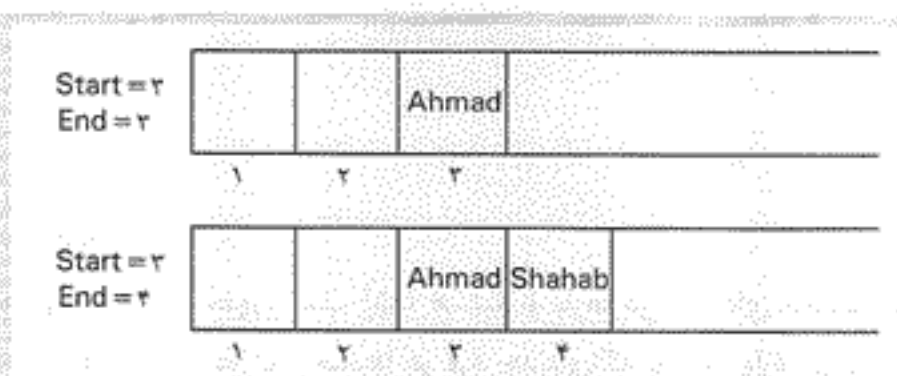
	Reza	Ahmad	
--	------	-------	--

۱

۲

۳

واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷



تمرین: الگوریتمی بنویسید که با استفاده از مفهوم صف، مقدار $n!$ را محاسبه کند.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسایل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

خلاصه مطالب

- برای دستیابی به روش حل مسایل، شناسایی سه پارامتر: مقادیر معلوم، محاسبات و خواسته‌های مسأله (مجهولات) کار را آسان‌تر می‌کند.
- مقادیر معلوم مقادیری هستند که در اختیار مسأله قرار می‌گیرند.
- محاسبات، عملیاتی ریاضی یا منطقی هستند که روی داده‌ها و اطلاعات انجام می‌شوند.
- خواسته‌های مسأله مقادیری هستند که در نتیجه انجام محاسبات روی مقادیر معلوم به دست می‌آیند.
- الگوریتم مجموعه‌ای از دستورالعمل‌هاست که با زبان دقیق و قابل فهم به همراه جزئیات کافی و به صورت مرحله به مرحله اجرا می‌شوند، به گونه‌ای که هدف خاصی را دنبال می‌کنند و شروع و خاتمه آن‌ها نیز مشخص است.
- انواع دستورالعمل‌ها در الگوریتم عبارتند از: ورودی، محاسباتی، خروجی، شرطی و حلقه‌ها.
- دستورالعمل‌های ورودی برای دریافت داده‌ها و دستورالعمل‌های خروجی برای نمایش داده و اطلاعات خروجی یا پیام‌های موردنیاز استفاده می‌شوند.
- دستورالعمل‌های محاسباتی برای انجام عملیات ریاضی و محاسباتی استفاده می‌شوند.
- دستورالعمل‌های شرطی برای انجام مقایسه‌ها و کنترل روند اجرای الگوریتم استفاده می‌شوند.
- دستورالعمل‌های تکرار (حلقه‌ها) برای ایجاد تکرار اجرای دستورالعمل‌ها استفاده می‌شوند.
- عملگر، یک یا مجموعه عملیاتی است که در ماشین الگوریتم به صورت از پیش آماده تعریف شده است.
- عملگرهای ریاضی عبارتند از: جمع (+)، تفریق (-)، ضرب (x)، تقسیم (/)، باقیمانده تقسیم صحیح (Mod) و خارج قسمت تقسیم صحیح (۱).
- عملگرهای مقایسه‌ای برای ایجاد انواع شرط‌ها در دستورالعمل‌های شرطی استفاده می‌شوند.
- عملگرهای منطقی «و» و «یا» برای ایجاد ترکیب‌های گوناگون از شرط‌ها در دستورالعمل‌های شرطی استفاده می‌شوند.
- عملگر Mod باقیمانده تقسیم صحیح و عملگر \ خارج قسمت تقسیم صحیح را محاسبه می‌کنند.

استاندارد مهارت: رابطه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

آزمون نظری

۱- نتیجه اجرای الگوریتم زیر چیست؟

- ۱- شروع
- ۲- $SUM \leftarrow 0$
- ۳- $I \leftarrow 1$
- ۴- $SUM \leftarrow SUM + I$
- ۵- $I \leftarrow I + 1$
- ۶- اگر $I \leq 100$ آن‌گاه برو به مرحله ۴
- ۷- SUM را نمایش بده
- ۸- پایان

- الف- اعداد ۱ تا ۹۹ را نمایش می‌دهد.
- ب- مجموع اعداد ۰ تا ۹۹ را نمایش می‌دهد.
- ج- مجموع اعداد ۱ تا ۱۰۰ را نمایش می‌دهد.
- د- مجموع اعداد ۱ تا ۹۹ را نمایش می‌دهد.

۲- کدام گزینه در رابطه با حاصل الگوریتم زیر درست است؟

- ۱- شروع کن
- ۲- $I \leftarrow 2$
- ۳- I را نمایش بده
- ۴- $K \leftarrow I \times 2$
- ۵- $I \leftarrow I + 2$
- ۶- K را نمایش بده
- ۷- اگر $I \leq 10$ آن‌گاه برو به مرحله ۵ در غیر این صورت برو به مرحله ۸
- ۸- پایان

- الف- ۲ و ۴ و ۴ و ۴ و ۴
- ب- ۲ و ۴ و ۶ و ۸
- ج- ۲ و ۴ و ۶ و ۸ و ۱۰
- د- ۲ و ۴ و ۴ و ۴ و ۴

۳- نتیجه اجرای الگوریتم زیر چیست؟

- ۱- شروع
- ۲- $I \leftarrow 1$
- ۳- اگر $I \bmod 2 = 0$ آن‌گاه I را نمایش بده

استاندارد مهارت: رایانه کار مرحله ۱	بیمه مهارتی: الگوریتم و فلوچارت	واحد کار: حل مسائل و طراحی الگوریتم مناسب برای آن‌ها
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۱

$$I \leftarrow I + 1$$

۵- اگر $I < 100$ آن‌گاه برو به مرحله ۳

۶- پایان

الف- اعداد بین صفر و ۱۰۰

ب- اعداد طبیعی فرد کوچک‌تر از ۱۰۰

ج- اعداد طبیعی زوج کوچک‌تر از ۱۰۰

د- اعداد بین ۱ و ۱۰۰

۴- کدام عملگر از اولویت بالاتری برخوردار است؟

الف- +

ب- /

ج- Mod

د- -

۵- در صورتی که $x = 5$ ، $y = -10$ و $z = 27$ باشد، نتیجه عبارت زیر چیست؟

$(x > 0)$ یا $(y < -20)$ و $(z = x)$

ب- نادرست

الف- درست

ج- ۱-

د- ۰

۶- الگوریتمی بنویسید که مضارب کوچک‌تر از ۱۰۰۰ عدد ۳ را محاسبه کرده و نمایش دهد.

۷- الگوریتمی بنویسید که حاصل جمع‌های زیر را محاسبه کرده و نمایش دهد.

$$1 + x + x^2 + \dots + x^n \quad (n \geq 0) \quad ۱-$$

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots + \frac{n}{2^n} \quad (n \geq 1) \quad ۲-$$

۸- الگوریتمی بنویسید که حاصل ضرب اعداد زوج بین دو عدد طبیعی و دلخواه M و N را محاسبه کرده و نمایش دهد.

۹- الگوریتمی بنویسید که ضرایب یک معادله درجه دوم را دریافت کرده و ریشه‌های آن را محاسبه نموده و نمایش دهد.

۱۰- الگوریتمی بنویسید که نام، نام خانوادگی و نمرات ۱۰ درس ۵ دانش‌آموز را دریافت و در آرایه ذخیره کند و سپس کارنامه هر دانش‌آموز را به صورت جداگانه نمایش دهد.

واحد کار حل مسائل و طراحی الگوریتم مناسب برای آن‌ها	بیمانه مهارتی الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۳۱	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

۱۱- الگوریتمی بنویسید که عدد طبیعی و دلخواه n را دریافت کرده و ارقام آن را جابه‌جا کند (مثلاً عدد ۲۴۸۵ به ۵۸۴۲ تبدیل شود).

۱۲- الگوریتمی بنویسید که مقدار زمان را براساس ثانیه دریافت کرده و مقدار ساعت‌ها، دقیقه‌ها و ثانیه‌های آن را نیز محاسبه کرده و نمایش دهد.

۱۳- الگوریتمی بنویسید که هزینه حمل کالا را براساس جدول زیر و با توجه به وزن کالا و روش حمل و نقل آن محاسبه کرده و نمایش دهد.

روش حمل و نقل	هزینه هر کیلوگرم کالا در مسافت یک کیلومتر
کامیون	۱۰۰۰ ریال
قطار	۱۴۰۰ ریال
کشتی	۱۲۰۰ ریال

۱۴- الگوریتمی بنویسید که عدد طبیعی و دلخواه m را دریافت کرده و اعداد کوچکتر از آن را که رقم یکان آن‌ها زوج و رقم صدگان آن‌ها فرد است، نمایش دهد.

مختص رسته برنامه‌نویسی

۱۵- زیرالگوریتمی بنویسید که یک آرایه را دریافت کرده سپس آن را با روش بازگشتی و به صورت حبابی مرتب کند.



هدف جزئی



توانایی ترسیم فلوجارت

زمان (ساعت)	
نظری	عملی
۱۰	۱۰

هدفهای رفتاری

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱- مفهوم فلوجارت را توضیح دهد.
- ۲- علایم ترسیم فلوجارت و نحوه استفاده از آن‌ها را توضیح دهد.
- ۳- بتواند انواع فلوجارت را برای مسایل متفاوت طراحی نماید.



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

کلیات

همان‌طور که می‌دانید در آرایه الگوریتم‌ها به منظور حل مسایل مختلف از جملات فارسی (یا سایر زبان‌ها) به همراه تعدادی از علائم قراردادی استفاده می‌شود، البته در الگوریتم‌های ساده و کوچک این روش به خوبی شما را برای تبدیل الگوریتم به برنامه موردنظر کمک می‌کند اما با پیچیده شدن الگوریتم و افزایش دستورالعمل‌ها این کار دشوار خواهد شد، بنابراین لازم است از روش‌های مفیدتری استفاده کنید؛ یکی از این روش‌ها ترسیم و طراحی فلوچارت است. فلوچارت (نمودار گردشی) در واقع مجموعه‌ای از اشکال و ترسیمات قراردادی است که دستورالعمل‌ها و ترتیب اجرای آن‌ها را مطابق با الگوریتم موردنظر نمایش می‌دهد. فلوچارت تمامی ویژگی‌های الگوریتم را داشته، علاوه بر این امکان درک بهتر از نحوه اجرای دستورالعمل‌ها را نیز به وجود می‌آورد.

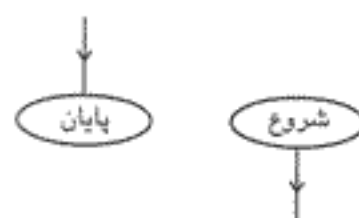
فلوچارت‌ها را می‌توانید پس از طراحی الگوریتم یا به‌طور مستقل پس از بررسی مسأله و انتخاب روش حل آن ترسیم کنید.

۲-۱-۱ علائم و اشکال در فلوچارت

برای تبدیل یک الگوریتم به فلوچارت از علائم قراردادی به جای استفاده از جملات استفاده می‌شود و به جای هر یک از دستورالعمل‌ها می‌توانید شکل و علامت معادل آن را به کار بگیرید و براساس ترتیب اجرای دستورالعمل‌ها این اشکال را با استفاده از خطوط فلش‌دار به یکدیگر متصل نمایید. لازم به ذکر است که تعداد فلش‌هایی که می‌تواند به هر علامت وارد شود، نامحدود است اما در تمام علائم به جز علامت شرط فقط یک فلش می‌تواند خارج شود.

۲-۱-۱-۱ علائم شروع و پایان

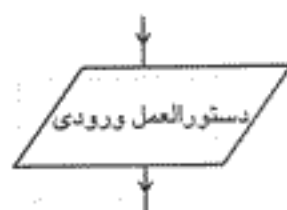
برای دستورالعمل‌های شروع کن و پایان از علامت بیضی به صورت زیر استفاده کنید:



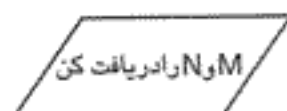
۲-۱-۲ علائم ورودی

برای دستورالعمل‌های ورودی مانند دریافت کن، بگیر و نظایر آن می‌توانید از شکل متوازی‌الاضلاع استفاده کنید.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

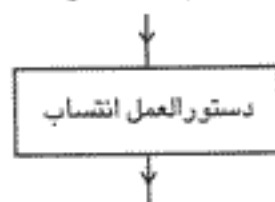


مثال:

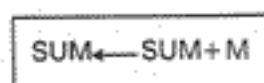


۳-۱-۲ علامت انتساب

برای دستورالعمل‌های محاسباتی از شکل مستطیل استفاده کنید.

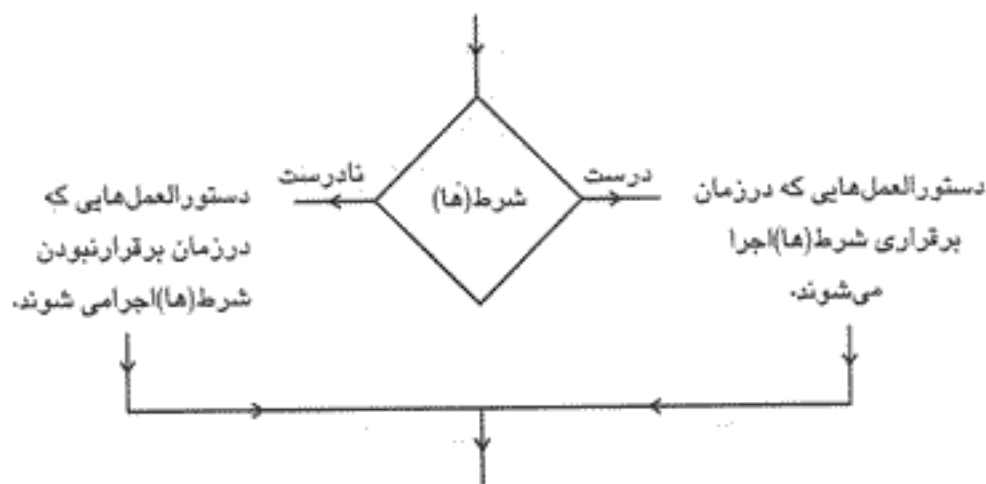


مثال:

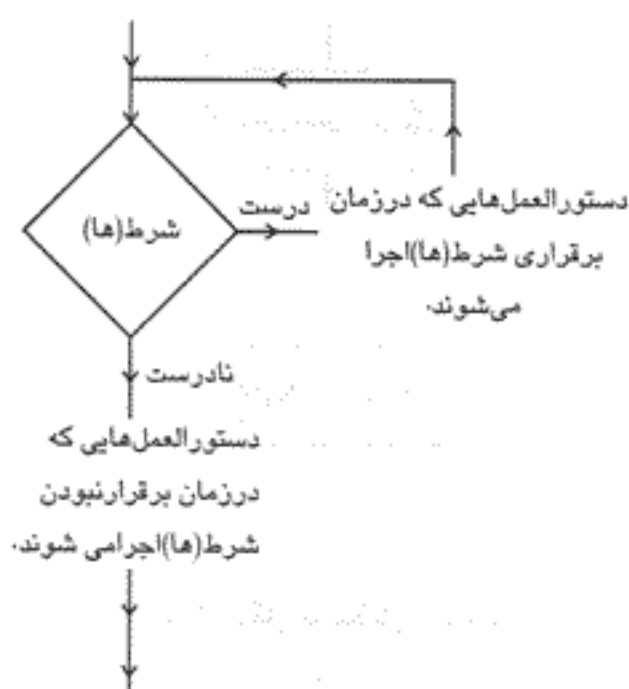


۴-۱-۲ علامت شرط

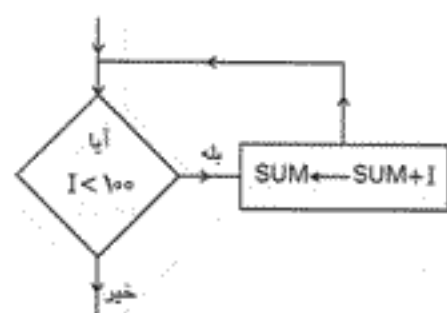
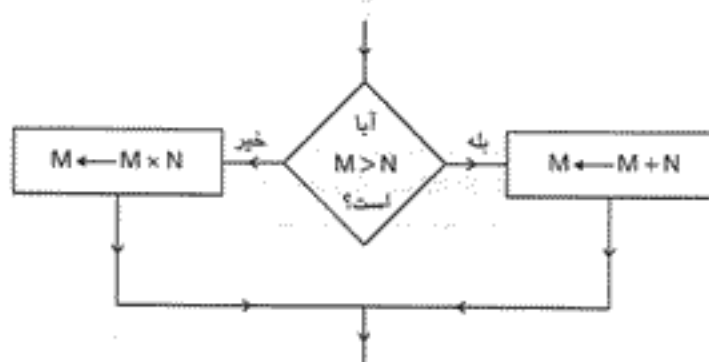
این علامت می‌تواند جایگزین دستورالعمل‌های شرطی شود و به یکی از صورت‌های زیر قابل استفاده است.



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



مثال:



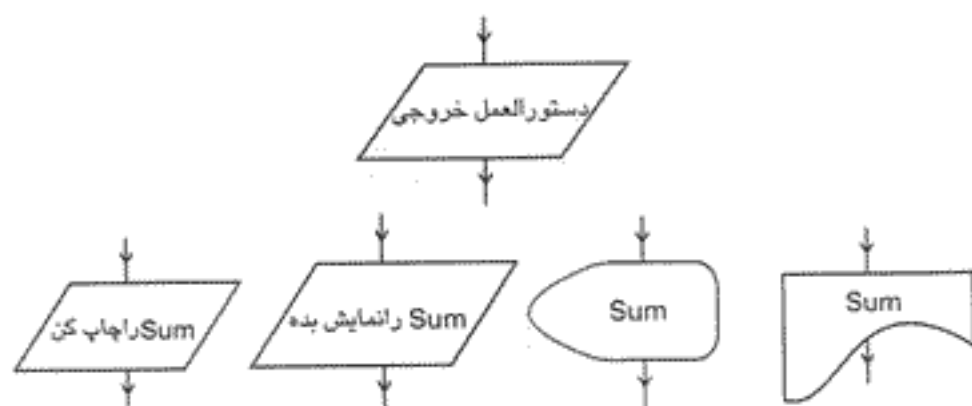
۵-۱-۲ علایم خروجی

با توجه به این که اطلاعات خروجی می‌توانند به چاپگر یا صفحه نمایش ارسال شوند از این علایم استفاده می‌شود:

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

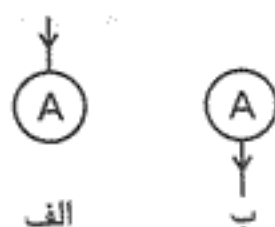


به علاوه می‌توانید از علامت متوازی‌الاضلاع به جای دستورالعمل خروجی نیز استفاده کنید.



۶-۱-۲ علامت اتصال

گاهی اوقات ممکن است فلوچارت به اندازه‌ای بزرگ باشد که در یک صفحه کاغذ قرار نگیرد؛ در چنین شرایطی می‌توانید از علامت اتصال استفاده کنید (شکل الف) و ادامه فلوچارت را در صفحه دیگری که آن هم با یک علامت اتصال دیگر شروع می‌شود (شکل ب)، ادامه دهید و در داخل هر دو علامت یک حرف از حروف الفبا یا یک عدد مثبت قرار دهید.



مثال ۱: الگوریتم و فلوچارتی بنویسید که عدد طبیعی و دلخواه M را دریافت کرده و زوج یا فرد بودن آن را معین کند.

۱- شروع

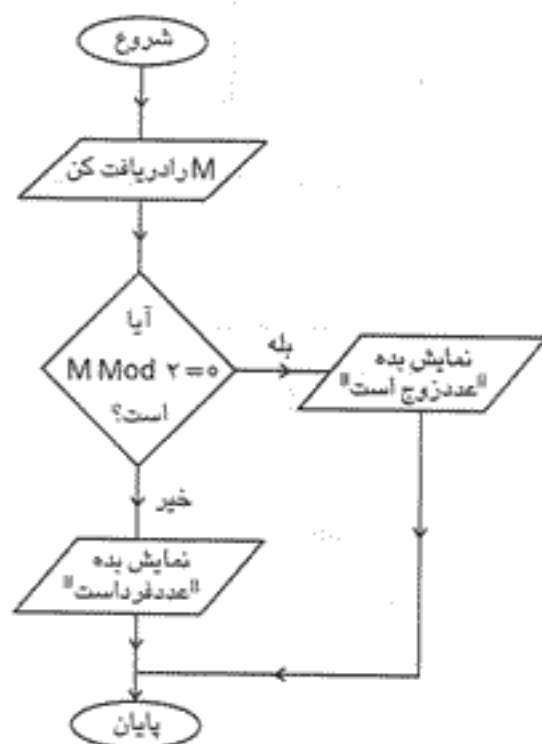
۲- M را دریافت کن

۳- اگر $M \bmod 2 = 0$ آن‌گاه "عدد زوج است" را نمایش بده در غیراین صورت "عدد فرد است" را نمایش بده

۴- پایان

واحد کار: ترسیم فلوچارت	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

فلوچارت الگوریتم قبل به این صورت خواهد بود:



تمرین: فلوچارتی رسم کنید که تعدادی عدد دلخواه را دریافت کرده و بزرگ‌ترین و کوچک‌ترین مقدار را محاسبه کند و نمایش دهد.

مثال ۲: الگوریتم و فلوچارتی بنویسید که عدد طبیعی و دلخواه N را دریافت کرده و کوچک‌ترین رقم آن را تعیین کند و نمایش دهد.

۱- شروع

۲- N را دریافت کن

۳- $R \leftarrow 0$

۴- $Min \leftarrow N \text{ Mod } 10$

۵- $R \leftarrow N \text{ Mod } 10$

۶- اگر $Min > R$ آن‌گاه $Min \leftarrow R$

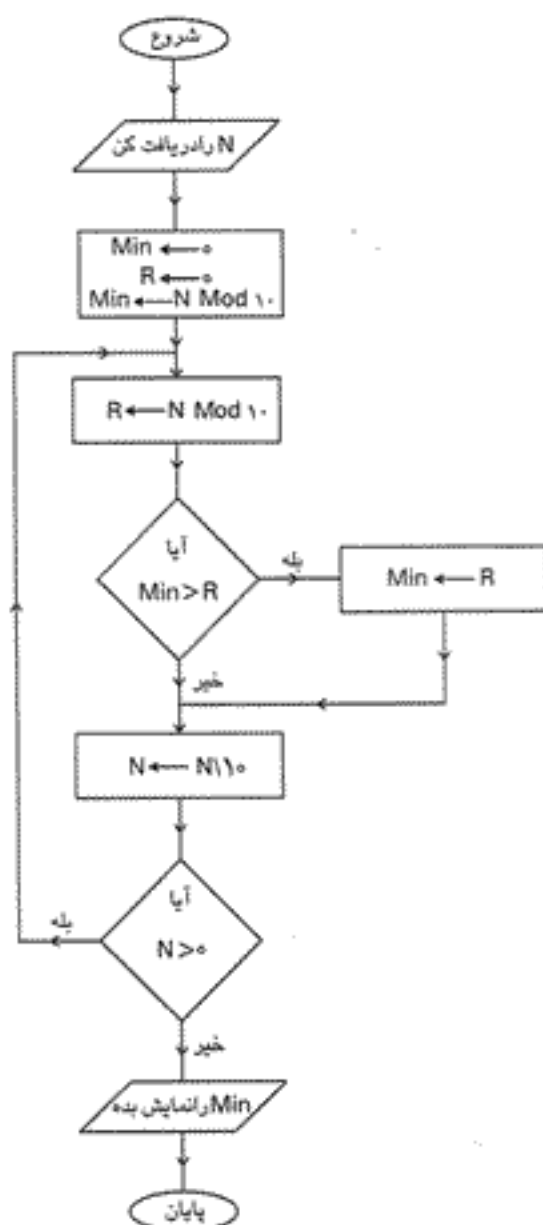
۷- $N \leftarrow N \setminus 10$

۸- اگر $N > 0$ آن‌گاه برو به مرحله ۵

۹- Min را نمایش بده

۱۰- پایان

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: ترسیم فلوجارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



تمرین: فلوجارتی رسم کنید که دو عدد را دریافت کرده و بخش‌پذیری عدد اول را بر عدد دوم بررسی کند.

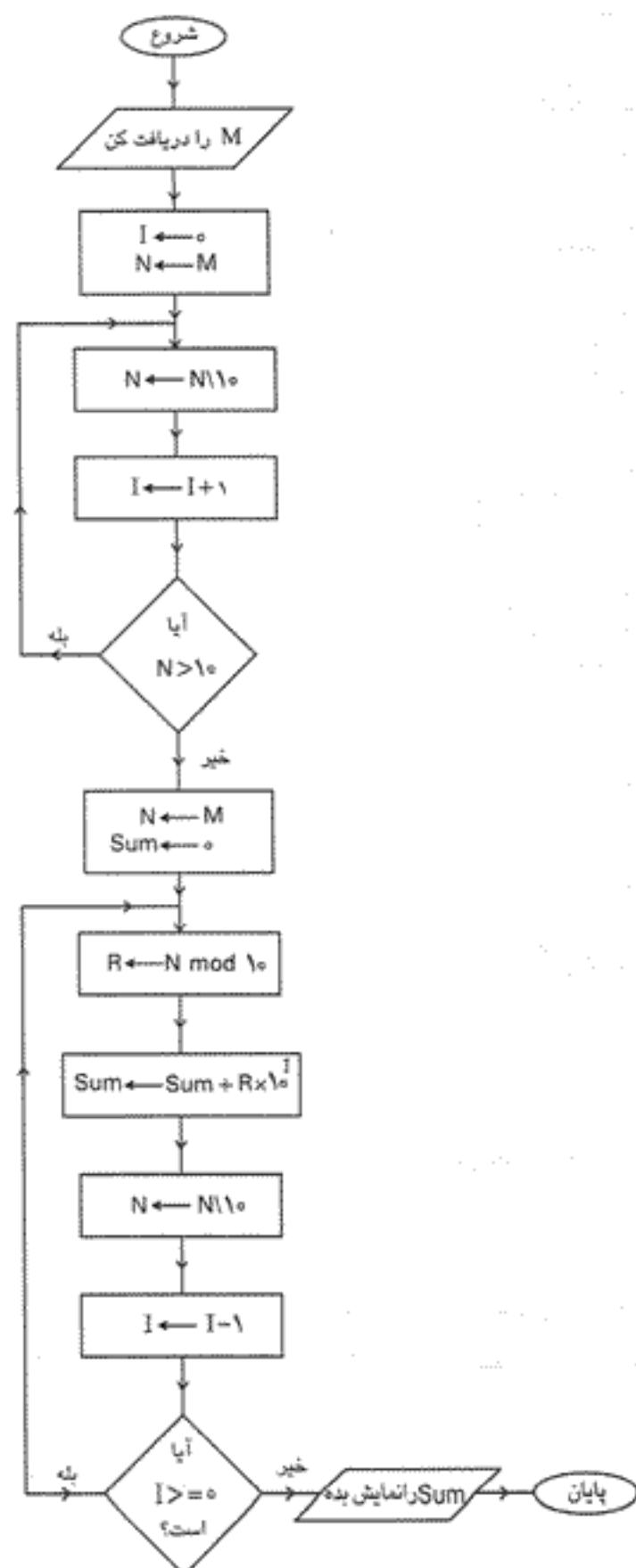
مثال ۳: الگوریتم و فلوجارتی بنویسید که عدد طبیعی و دلخواه M را دریافت کرده و به صورت برعکس نمایش دهد، به عنوان مثال عدد ۵۲۷۴ را به صورت ۴۷۲۵ نمایش دهد.

۱- شروع

۲- M را دریافت کن

۳- $N \leftarrow M$ و $I \leftarrow 0$

واحد کار: ترسیم فلوچارت	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷



۴- $N \leftarrow N \setminus 10$

۵- $I \leftarrow I + 1$

۶- اگر $N > 10$ آن گاه برو به مرحله ۴

۷- $N \leftarrow M$ و $Sum \leftarrow 0$

۸- $R \leftarrow N \bmod 10$

۹- $Sum \leftarrow Sum + R \times 10^I$

۱۰- $N \leftarrow N \setminus 10$

۱۱- $I \leftarrow I - 1$

۱۲- اگر $I \geq 0$ آن گاه برو به مرحله ۸

۱۳- Sum را نمایش بده

۱۴- پایان

استاندارد مهارت: رایانه کار درجه ۱	پسمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

تمرین: فلوچارتی رسم کنید که مجموع و تعداد اعداد طبیعی موجود بین اعداد طبیعی و دلخواه M و N را نمایش دهد ($M < N$).

مثال ۴: الگوریتم و فلوچارتی بنویسید که عدد طبیعی و دلخواه m را دریافت کرده و مقسوم‌علیه‌های زوج آن را نمایش دهد.

۱- شروع

۲- m را دریافت کن

۳- $k \leftarrow 1$

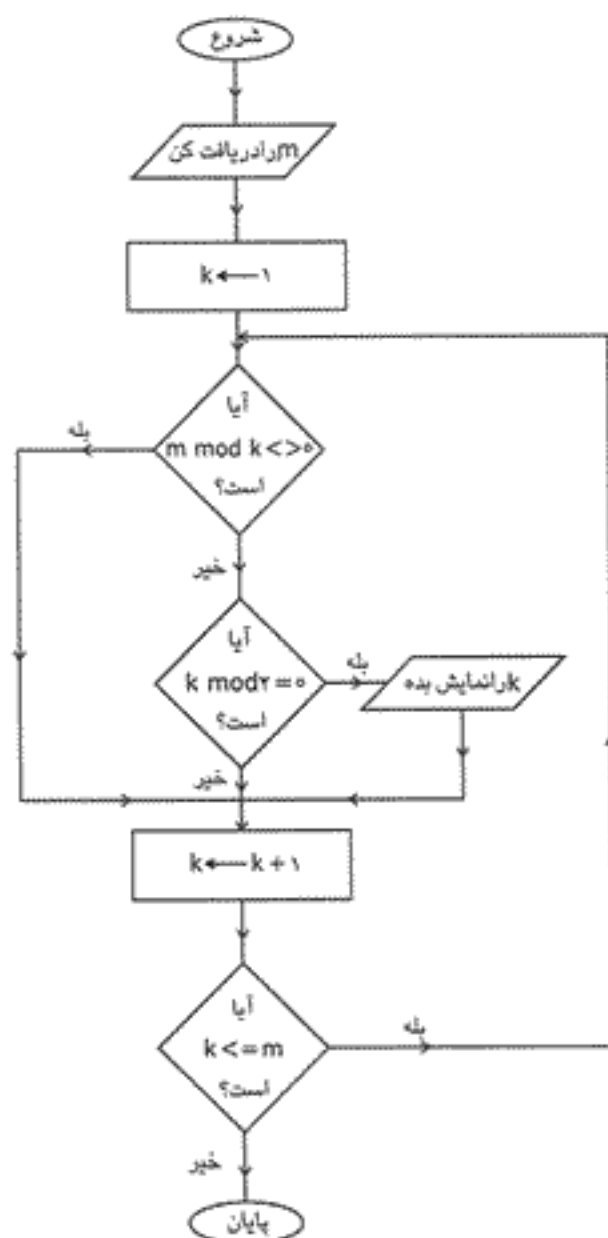
۴- اگر $(m \bmod k) \neq 0$ آن‌گاه برو به مرحله ۶

۵- اگر $(k \bmod 2) = 0$ آن‌گاه k را نمایش بده

۶- $k \leftarrow k + 1$

۷- اگر $k \leq m$ آن‌گاه برو به مرحله ۴

۸- پایان



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

تمرین: فلوچارتی رسم کنید که عدد طبیعی و دلخواه N را دریافت کرده و مقدار $N!$ را محاسبه کند و نمایش دهد.

مثال ۵: الگوریتم و فلوچارتی بنویسید که مجموع زیر را محاسبه کرده و نمایش دهد ($n \geq 0$ و n تعداد جملات است).
 $Sum = 1 + 3 + 5 + \dots + (2n + 1)$

۱- شروع

۲- n را دریافت کن

۳- $Sum \leftarrow 0$ و $I \leftarrow 1$ و $K \leftarrow 0$

۴- $Sum \leftarrow Sum + I$

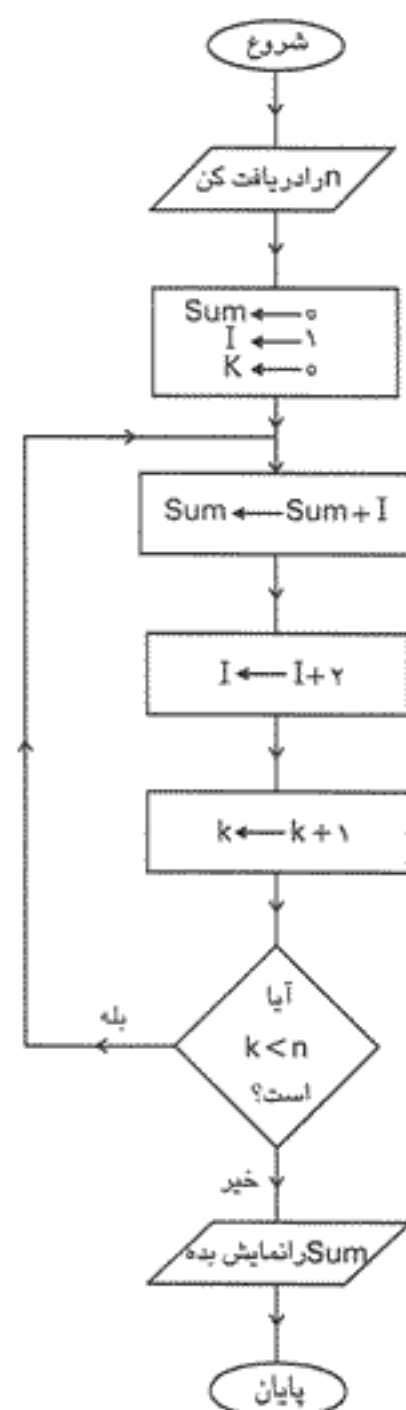
۵- $I \leftarrow I + 2$

۶- $k \leftarrow k + 1$

۷- اگر $k < n$ آن گاه برو به مرحله ۴

۸- Sum را نمایش بده

۹- پایان



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: ترسیم فلوجارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

مثال ۶: الگوریتم و فلوجارتی بنویسید که مجموع زیر را محاسبه کرده و نمایش دهد ($n \geq 1$ و n تعداد جملات است).

$$Sum = \frac{2}{x^2} + \frac{4}{x^4} + \frac{6}{x^6} + \dots + \frac{2n}{x^{2n}}$$

۱- شروع

۲- x و n را دریافت کن

۳- $Sum \leftarrow 0$ و $I \leftarrow 2$ و $k \leftarrow 0$

۴- $Sum \leftarrow Sum + \frac{I}{x^I}$

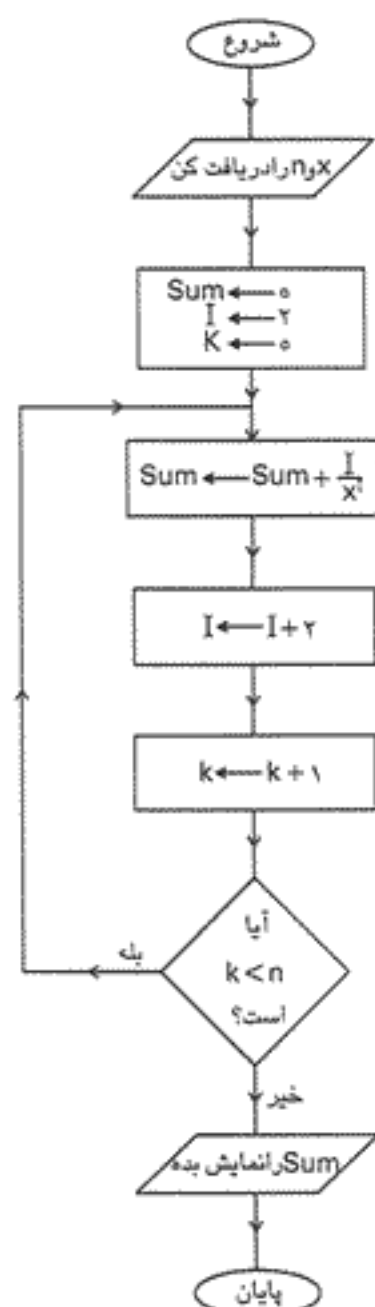
۵- $I \leftarrow I + 2$

۶- $k \leftarrow k + 1$

۷- اگر $k < n$ آن گاه برو به مرحله ۴

۸- Sum را نمایش بده

۹- پایان



واحد کار: ترسیم فلوچارت	بیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

تمرین: فلوچارتی رسم کنید که یک عدد طبیعی در مبنای ۱۶ را دریافت کند و معادل آن را در مبنای ۱۰ محاسبه کرده و نمایش دهد.

مثال ۷: الگوریتم و فلوچارتی بنویسید که مجموع زیر را محاسبه کرده و نمایش دهد ($n \geq 0$).

$$\text{Sum} = \frac{3}{2!} + \frac{4}{4!} + \frac{5}{6!} + \dots + \frac{(n+3)}{(2(n+1))!}$$

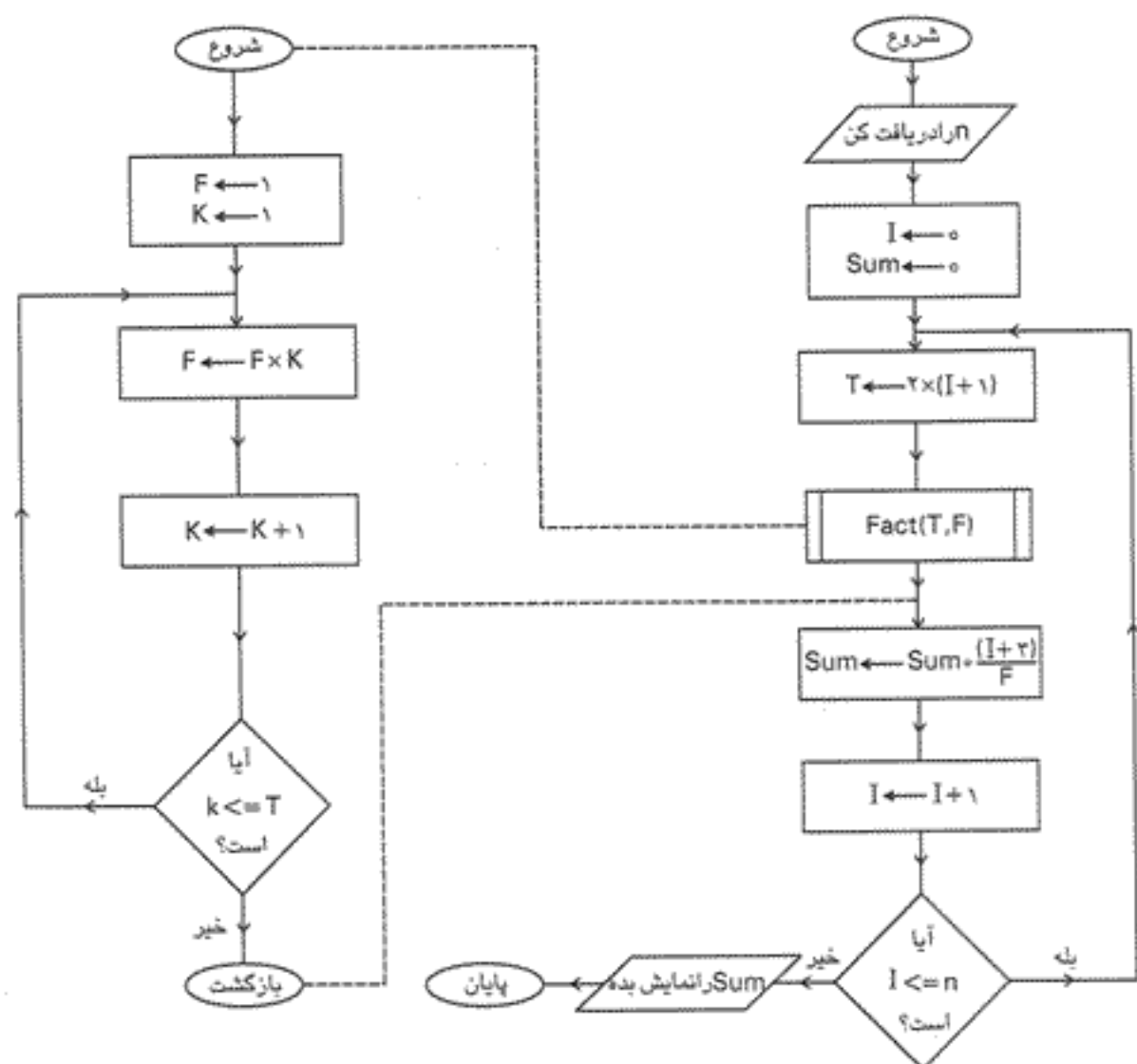
الگوریتم اصلی

- ۱- شروع
- ۲- n را دریافت کن
- ۳- $I \leftarrow 0$ و $\text{Sum} \leftarrow 0$
- ۴- $T \leftarrow 2 \times (I+1)$
- ۵- زیرالگوریتم $\text{Fact}(T)$ و F را اجرا کن
- ۶- $\text{Sum} \leftarrow \text{Sum} + \frac{(I+3)}{F}$
- ۷- $I \leftarrow I+1$
- ۸- اگر $I \leq n$ آن گاه برو به مرحله ۴
- ۹- Sum را نمایش بده
- ۱۰- پایان

زیرالگوریتم $\text{Fact}(T)$ و F

- ۱- شروع
- ۲- $F \leftarrow 1$ و $K \leftarrow 1$
- ۳- $F \leftarrow F \times K$
- ۴- $K \leftarrow K+1$
- ۵- اگر $K \leq T$ آن گاه برو به مرحله ۳
- ۶- بازگشت

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوجارت	واحد کار: ترسیم فلوجارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



تمرین: فلوجارتی رسم کنید که مجموع زیر را محاسبه کرده و نمایش دهد (با فرض آن که $n \geq 0$).

$$\text{Sum} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^{(2n+1)}}{(2n+1)!}$$

مثال ۸: الگوریتم و فلوجارتی بنویسید که n عدد طبیعی و دلخواه را به صورت یک آرایه دریافت کرده و عددی که بیشترین تکرار را دارد، محاسبه کرده و عدد و تعداد دفعات تکرار آن را نمایش دهد.

۱- شروع

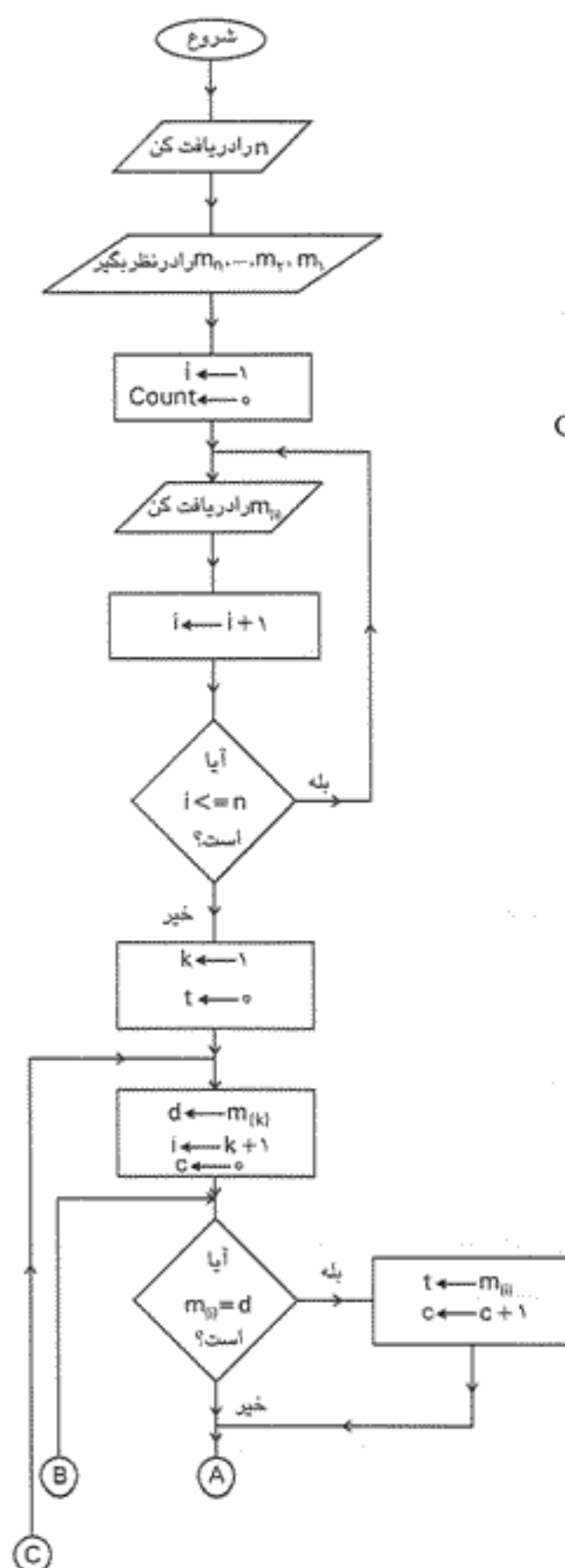
۲- n را دریافت کن

۳- m_1 و m_2 و ... و m_n را در نظر بگیر

۴- $i \leftarrow 1$ و $\text{Count} \leftarrow 0$

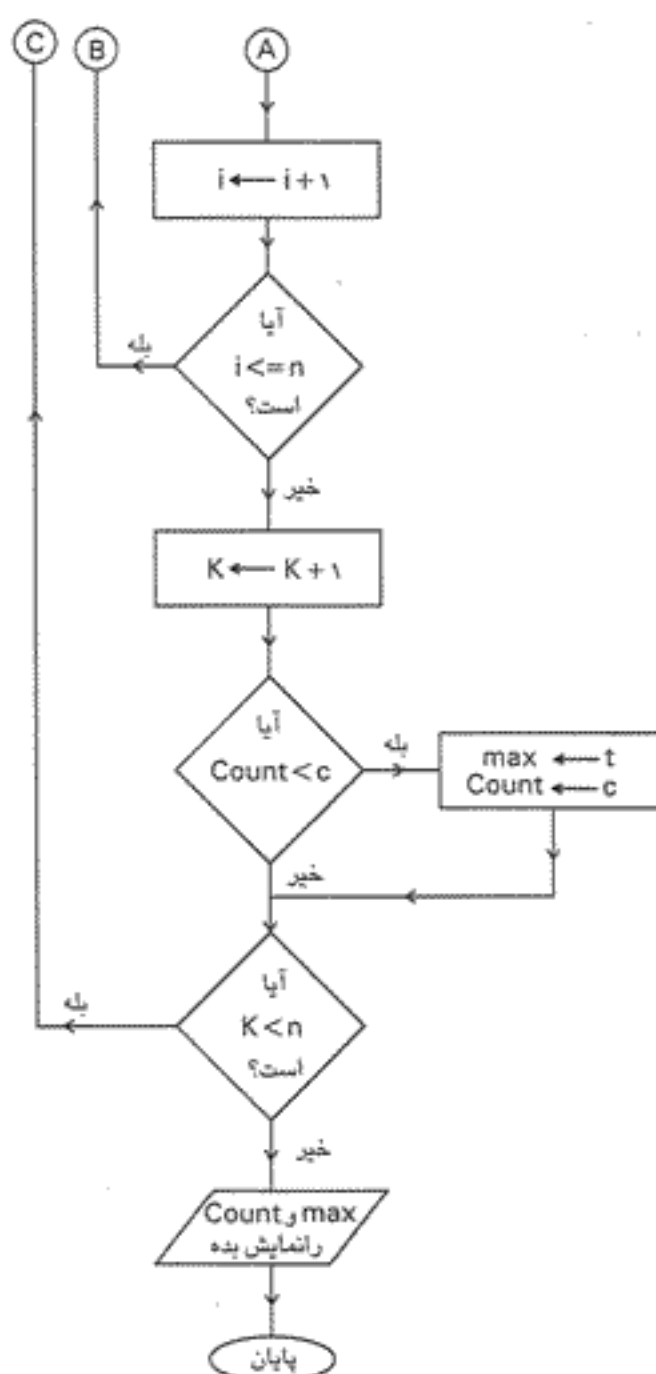
۵- $m_{(i)}$ را دریافت کن

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۲۲/۲۷	شماره شناسایی: ۳-۲۲/۲۷-۲	شماره شناسایی: ۳-۲۲/۲۷-۲۲



- ۶- $i \leftarrow i + 1$
- ۷- اگر $i \leq n$ آن گاه برو به مرحله ۵
- ۸- $t \leftarrow 0$ و $k \leftarrow 1$
- ۹- $c \leftarrow 0$ و $i \leftarrow k + 1$ و $d \leftarrow m_{(k)}$
- ۱۰- اگر $m_{(i)} = d$ آن گاه $t \leftarrow m_{(i)}$ و $c \leftarrow c + 1$
- ۱۱- $i \leftarrow i + 1$
- ۱۲- اگر $i \leq n$ آن گاه برو به مرحله ۱۰
- ۱۳- $k \leftarrow k + 1$
- ۱۴- اگر $\text{Count} < c$ آن گاه $\text{max} \leftarrow t$ و $\text{Count} \leftarrow c$
- ۱۵- اگر $k < n$ آن گاه برو به مرحله ۹
- ۱۶- max و Count را نمایش بده
- ۱۷- پایان

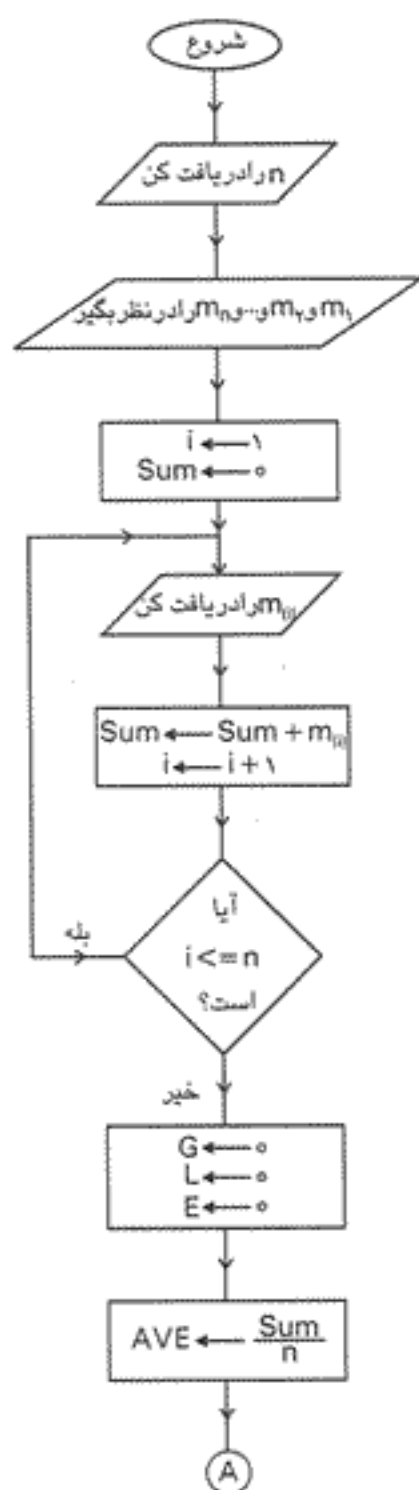
استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



تمرین: فلوچارتی رسم کنید که تعدادی عدد را به صورت یک آرایه دریافت کرده و بزرگ‌ترین و کوچک‌ترین عدد را در بین آن‌ها محاسبه کند و نمایش دهد.

مثال ۹: الگوریتم و فلوچارتی بنویسید که n عدد دلخواه را به صورت یک آرایه دریافت کرده و تعیین کند چند عدد از میانگین آن‌ها بزرگ‌تر و چند عدد از میانگین آن‌ها کوچک‌تر و چند عدد با میانگین آن‌ها مساوی است.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



۱- شروع

۲- n را دریافت کن

۳- m_1 و m_2 و ... و m_n را دریافت بگیر

۴- $i \leftarrow 1$ و $Sum \leftarrow 0$

۵- $m_{(i)}$ را دریافت کن

۶- $Sum \leftarrow Sum + m_{(i)}$ و $i \leftarrow i + 1$

۷- اگر $i \leq n$ آن گاه برو به مرحله ۵

۸- $G \leftarrow 0$ و $L \leftarrow 0$ و $E \leftarrow 0$

۹- $AVE \leftarrow \frac{Sum}{n}$

۱۰- $i \leftarrow 1$

۱۱- اگر $m_{(i)} > AVE$ آن گاه $G \leftarrow G + 1$

۱۲- اگر $m_{(i)} < AVE$ آن گاه $L \leftarrow L + 1$

۱۳- اگر $m_{(i)} = AVE$ آن گاه $E \leftarrow E + 1$

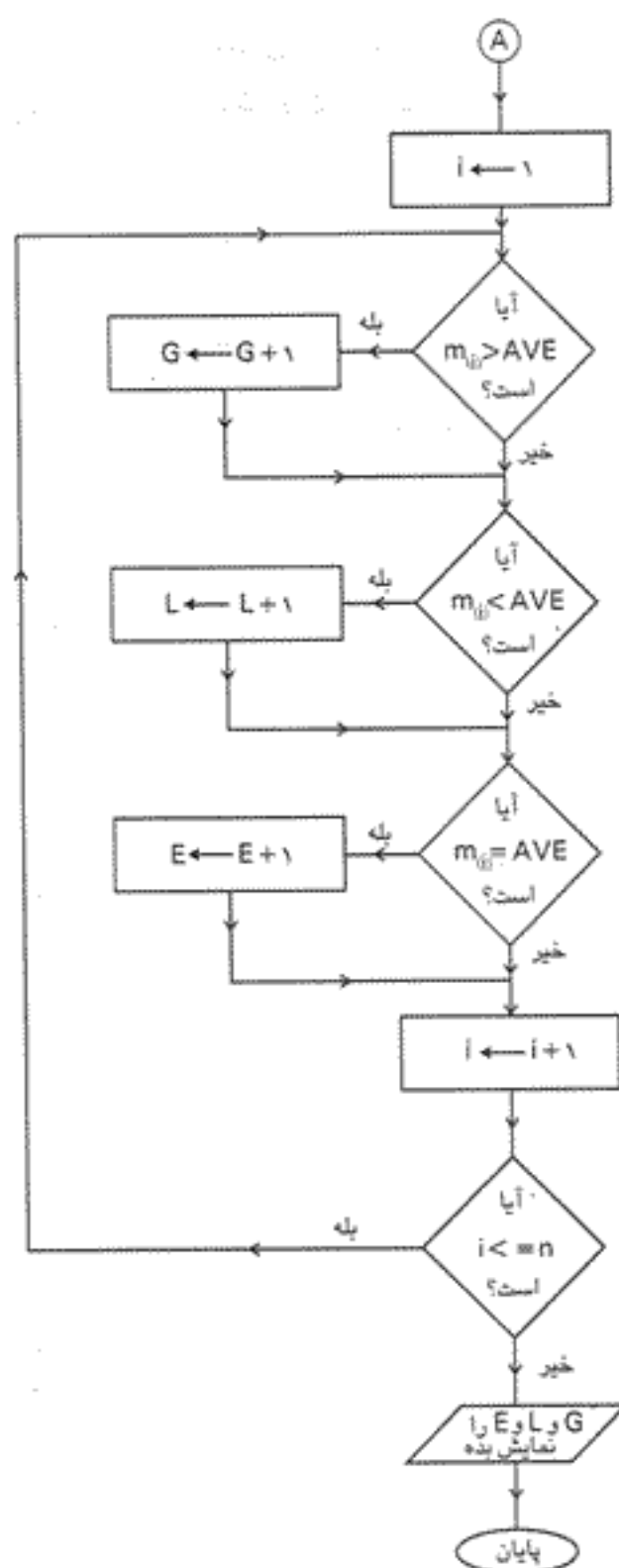
۱۴- $i \leftarrow i + 1$

۱۵- اگر $i \leq n$ آن گاه برو به مرحله ۱۱

۱۶- G و L و E را نمایش بده

۱۷- پایان

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



تمرین: فلوچارتی رسم کنید که دو آرایه یک بعدی را دریافت کرده و حاصل جمع آن‌ها را در آرایه یک بعدی جداگانه‌ای ذخیره نموده و نمایش دهد.

واحد کار: ترسیم فلوچارت	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

مثال ۱۰: الگوریتم و فلوچارتی بنویسید که اعضای یک ماتریس 3×4 را دریافت کرده، سپس مجموع هر سطر و ستون را در ماتریس‌های جداگانه‌ای ذخیره کرده و نمایش دهد.

۱- شروع

۲- $M_{1,1}$ و $M_{1,2}$... و $M_{3,4}$ را در نظر بگیر

۳- Row_1 و Row_2 و Row_3 را در نظر بگیر

۴- Col_1 و Col_2 و Col_3 و Col_4 را در نظر بگیر

۵- $i \leftarrow 1$

۶- $j \leftarrow 1$

۷- $M_{(i,j)}$ را دریافت کن

۸- $j \leftarrow j + 1$

۹- اگر $j \leq 4$ آن‌گاه برو به مرحله ۷

۱۰- $i \leftarrow i + 1$

۱۱- اگر $i \leq 3$ آن‌گاه برو به مرحله ۶

۱۲- $i \leftarrow 1$

۱۳- $j \leftarrow 1$ و $Sum \leftarrow 0$

۱۴- $Sum \leftarrow Sum + M_{(i,j)}$

۱۵- $j \leftarrow j + 1$

۱۶- اگر $j \leq 4$ آن‌گاه برو به مرحله ۱۴

۱۷- $Row_i \leftarrow Sum$

۱۸- $i \leftarrow i + 1$

۱۹- اگر $i \leq 3$ آن‌گاه برو به مرحله ۱۳

۲۰- $i \leftarrow 1$

۲۱- $j \leftarrow 1$ و $Sum \leftarrow 0$

۲۲- $Sum \leftarrow Sum + M_{(j,i)}$

۲۳- $j \leftarrow j + 1$

۲۴- اگر $j \leq 3$ آن‌گاه برو به مرحله ۲۲

۲۵- $Col_j \leftarrow Sum$

۲۶- $i \leftarrow i + 1$

۲۷- اگر $i \leq 4$ آن‌گاه برو به مرحله ۲۱

۲۸- $i \leftarrow 1$

۲۹- $j \leftarrow 1$

استاندارد مهارت: رایانه کار درجه ۱	بسمه مهارتی: الگوریتم و فلوجارت	واحد کار: ترسیم فلوجارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

۳۰- $M_{(i,j)}$ را نمایش بده

۳۱- $j \leftarrow j+1$

۳۲- اگر $j \leq 4$ آن گاه برو به مرحله ۳۰

۳۳- $i \leftarrow i+1$

۳۴- اگر $i \leq 3$ آن گاه برو به مرحله ۲۹

۳۵- $i \leftarrow 1$

۳۶- $Row_{(i)}$ را نمایش بده

۳۷- $i \leftarrow i+1$

۳۸- اگر $i \leq 3$ آن گاه برو به مرحله ۳۶

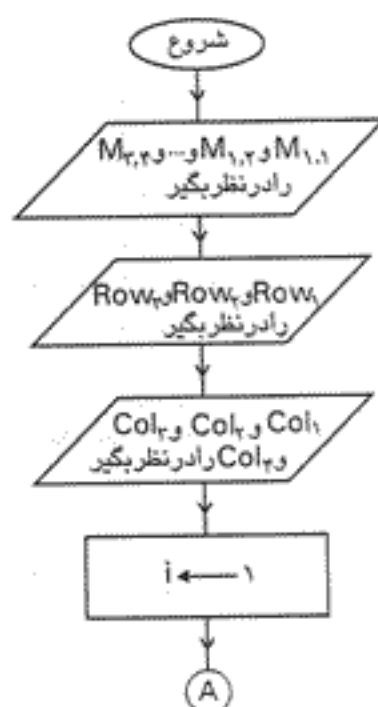
۳۹- $i \leftarrow 1$

۴۰- $Col_{(i)}$ را نمایش بده

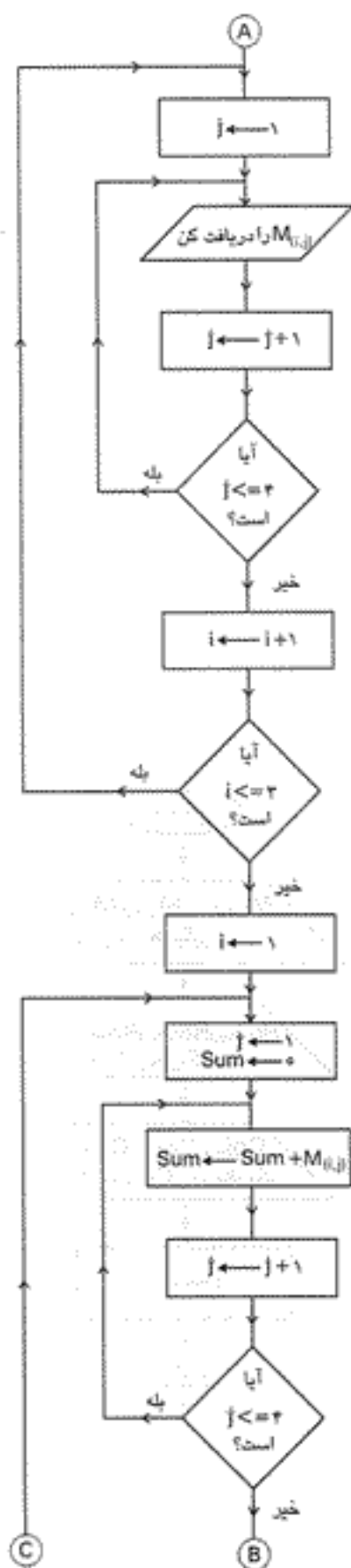
۴۱- $i \leftarrow i+1$

۴۲- اگر $i \leq 4$ آن گاه برو به مرحله ۴۰

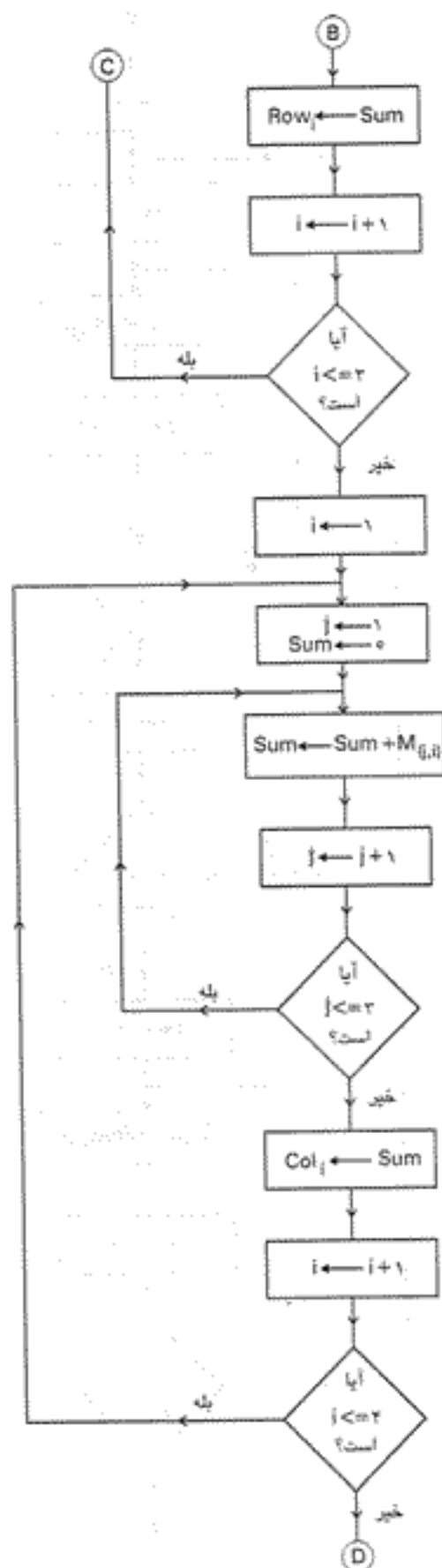
۴۳- پایان



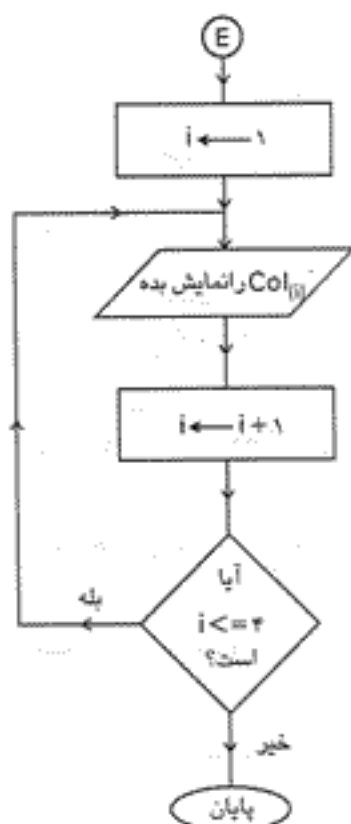
استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲



واحد کتار: ترسیم فلوچارت	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷



استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

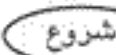


تمرین: فلوچارتی رسم کنید که یک آرایه دو بعدی ($m \times n$) را دریافت کند و مجموع سطرها و ستون‌های آن را محاسبه کرده، نمایش دهد.

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۲

خلاصه مطالب

- فلوچارت یا نمودار گردش مجموعه‌ای از اشکال و ترسیمات قراردادی است که دستورالعمل‌ها و ترتیب اجرای آن‌ها را نمایش می‌دهد.

- برای دستورالعمل شروع کن از علامت  شروع استفاده می‌شود.

- برای دستورالعمل پایان از علامت  پایان استفاده می‌شود.

- از علامت  برای دستورالعمل‌های ورودی و خروجی استفاده می‌شود.

- از علامت  برای دستورالعمل انتساب استفاده می‌شود.

- برای دستورالعمل شرطی از علامت  استفاده می‌شود.

- از علامت  برای نمایش اطلاعات روی صفحه نمایش استفاده می‌شود.

- از علامت  برای چاپ اطلاعات روی کاغذ استفاده می‌شود.

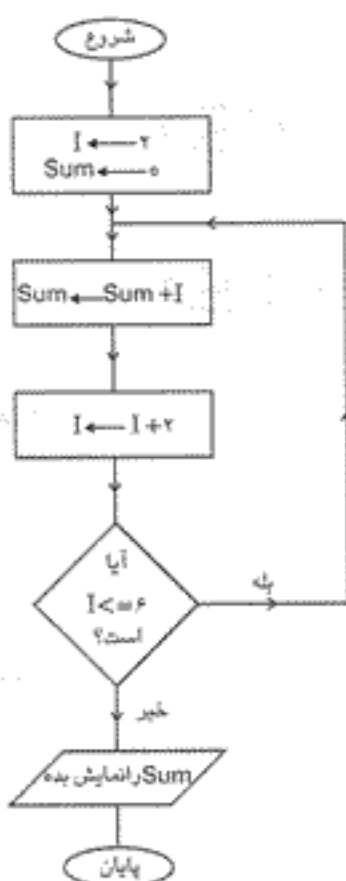
- از علامت  برای اتصال قسمت‌های مختلف یک فلوچارت به یکدیگر استفاده می‌شود.

واحد کار: ترسیم فلوچارت	بیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

آزمون نظری

۱- کدام گزینه در رابطه با اجرای فلوچارت زیر درست است؟

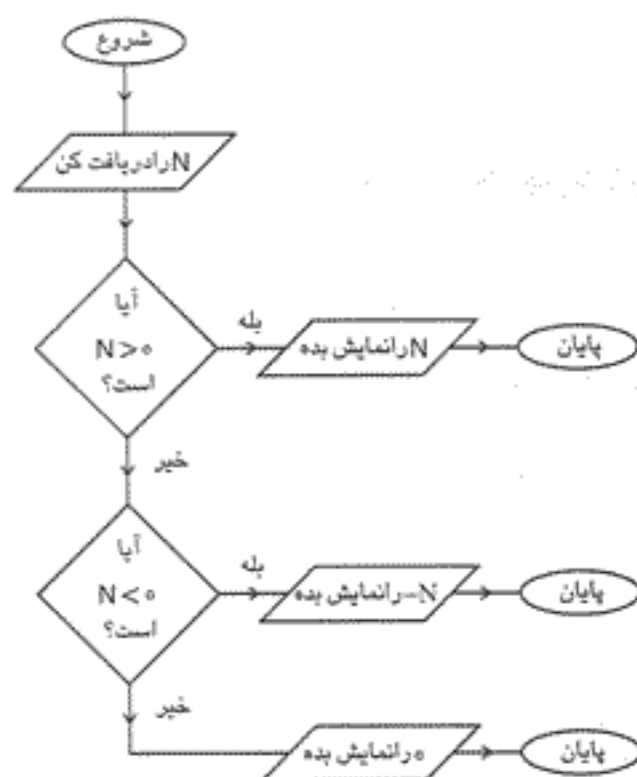
- الف- نمایش اعداد زوج کوچک‌تر از ۶
- ب- نمایش اعداد زوج مساوی ۶
- ج- نمایش مجموع اعداد زوج کوچک‌تر یا مساوی ۶
- د- مجموع اعداد زوج کوچک‌تر از ۶



۲- عملکرد فلوچارت بعد چیست؟

- الف- نمایش قرینه یک عدد
- ب- نمایش قدرمطلق یک عدد
- ج- نمایش عدد به صورت گرد شده
- د- نمایش حاصل ضرب هر عدد دلخواه در ۱-

استاندارد مهارت: رایانه کار درجه ۱	پیمانه مهارتی: الگوریتم و فلوچارت	واحد کار: ترسیم فلوچارت
شماره شناسایی: ۳-۴۲/۲۷	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷-۲۴



۳- خروجی فلوچارت زیر چیست؟

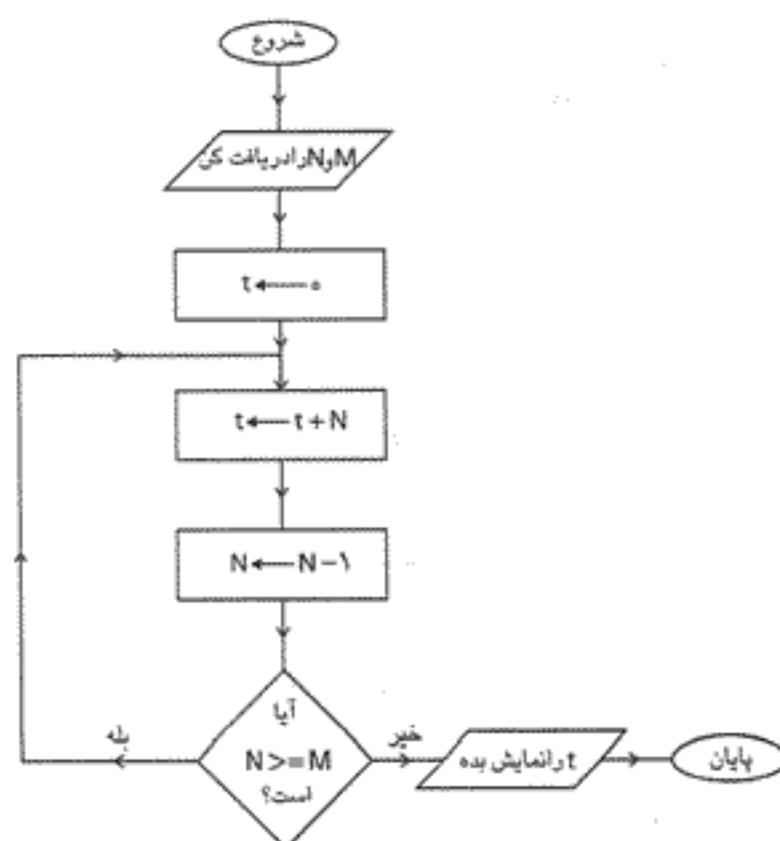
الف- اعداد بین N و M

ب- مجموع اعداد بین N و M

ج- حاصل ضرب N در M

د- مجموع N و M اعداد موجود

در بین آن‌ها $(N > M)$



واحد کار: ترسیم فلوچارت	پیمانه مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۴۲/۲۷-۲۲	شماره شناسایی: ۳-۴۲/۲۷-۲	شماره شناسایی: ۳-۴۲/۲۷

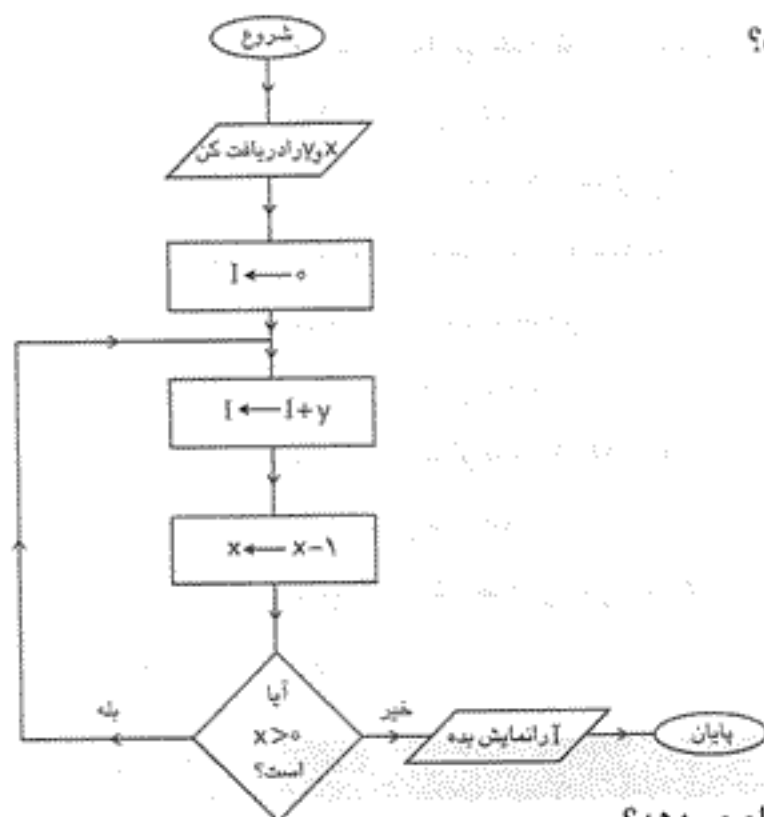
۴- خروجی این فلوچارت چیست؟

الف- حاصل جمع x و y

ب- حاصل ضرب x در y

ج- مجموع اعداد بین x و y

د- باقیمانده تقسیم y بر x



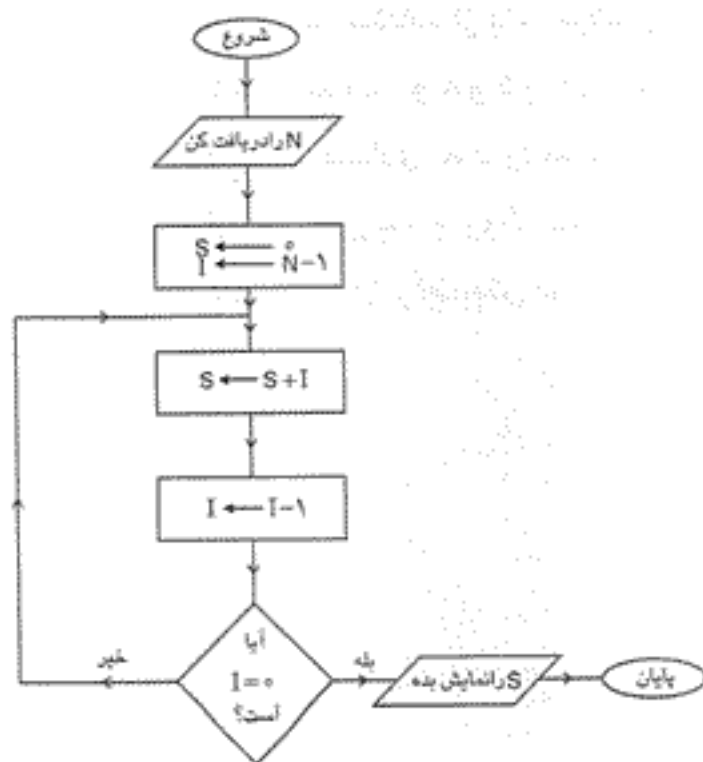
۵- فلوچارت بعد چه عملی را انجام می‌دهد؟

الف- محاسبه حاصل ضرب اعداد کوچک‌تر از N

ب- محاسبه حاصل جمع اعداد کوچک‌تر از N

ج- محاسبه مقدار $N!$

د- محاسبه مقدار N^1

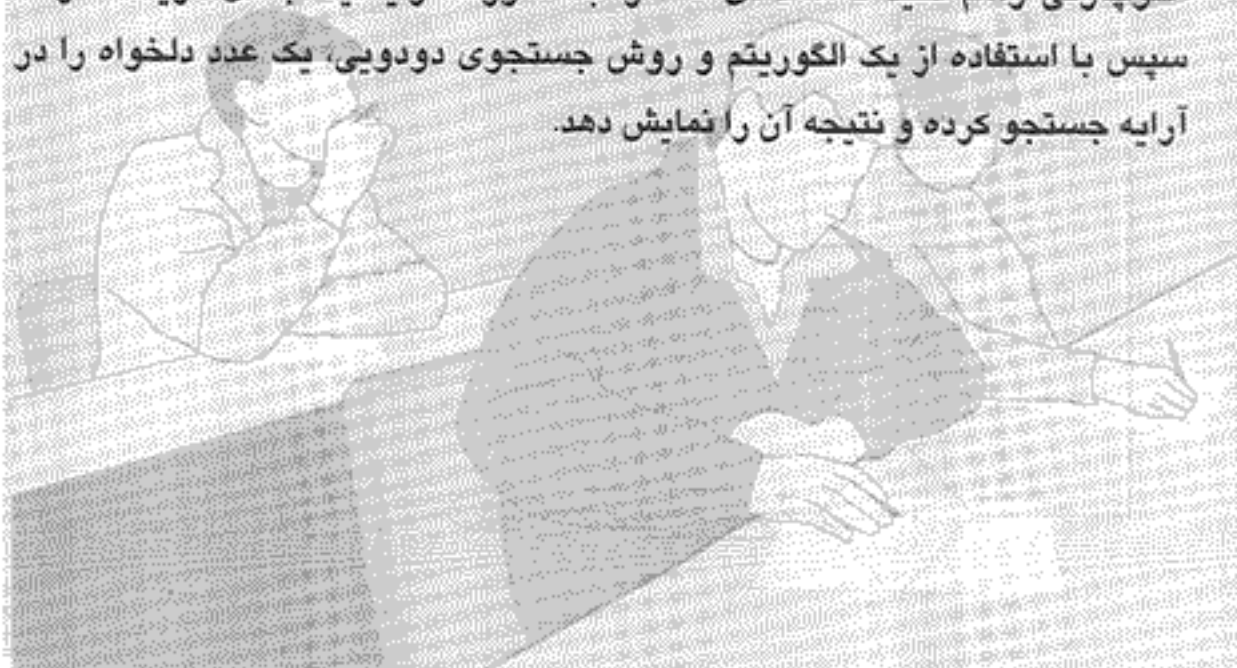


واحد کارا ترسیم فلوچارت	پشته مهارتی: الگوریتم و فلوچارت	استاندارد مهارت: رایانه کار درجه ۱
شماره شناسایی: ۳-۲۲/۲۷-۲۲	شماره شناسایی: ۳-۲۲/۲۷-۲	شماره شناسایی: ۳-۲۲/۲۷

- ۶- فلوچارتی رسم کنید که مضارب عدد ۵ را که کوچکتر از ۵۰۰ هستند، محاسبه کرده و به همراه مجموع آن‌ها نمایش دهد.
- ۷- فلوچارتی رسم کنید که دو آرایه یک بعدی را دریافت کرده و حاصل جمع آن‌ها را در آرایه یک بعدی جداگانه‌ای ذخیره کند و نمایش دهد.
- ۸- فلوچارتی رسم کنید که یک عدد طبیعی و دلخواه را دریافت کند و حاصل ضرب ارقام آن‌ها را محاسبه کرده و نمایش دهد.
- ۹- فلوچارتی رسم کنید که یک ماتریس $M \times N$ را دریافت کرده و کوچکترین و بزرگترین عضو و موقعیت آن را نمایش دهد.
- ۱۰- فلوچارتی رسم کنید که یک عدد را در مبنای ۱۰ دریافت کند و آن را به مبنای ۲ تبدیل کرده و نمایش دهد.

مخصوص رشته برنامه‌نویسی

- ۱۱- فلوچارتی رسم کنید که N عدد دلخواه را به صورت آرایه دریافت کرده و سپس آن را مرتب کند و نمایش دهد.
- ۱۲- فلوچارتی رسم کنید که با استفاده از یک زیرالگوریتم بازگشتی، دنباله فیبوناچی را محاسبه کرده و نمایش دهد (..... و ۸ و ۵ و ۳ و ۲ و ۱ و ۰).
- ۱۳- فلوچارتی رسم کنید که تعدادی عدد را به صورت آرایه یک بعدی دریافت کرده، سپس با استفاده از یک الگوریتم و روش جستجوی دودویی، یک عدد دلخواه را در آرایه جستجو کرده و نتیجه آن را نمایش دهد.



آزمون پایانی

۱- الگوریتم و فلوچارتی بنویسید که یک آرایه $m \times m$ را دریافت کرده و مجموع اعضای قطر اصلی و فرعی آن را محاسبه کند.

نکته: قطر فرعی در آرایه‌ای که تعداد سطرها و ستون‌های آن با هم برابر باشند، اعضای هستند که در موقعیت متناظر با اعضای قطر اصلی قرار دارند؛ به عنوان مثال برای آرایه زیر، قطر فرعی را مشاهده کنید.

$$M_{4 \times 4} = \begin{bmatrix} 4 & 3 & 7 & 9 \\ 12 & 0 & 1 & 14 \\ 5 & 6 & 18 & 25 \\ 47 & 12 & 9 & 73 \end{bmatrix}$$

قطر فرعی

۲- الگوریتم و فلوچارتی بنویسید که حاصل جمع زیر را با توجه به مقدار m محاسبه نماید. (m عدد طبیعی است).

$$\text{Sum} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{m}$$

۳- الگوریتم و فلوچارتی بنویسید که یک ماتریس $m \times m$ را دریافت کرده و اعضای روی قطر فرعی را با اعضای روی قطر اصلی عوض کند.

۴- الگوریتم و فلوچارتی بنویسید که عدد طبیعی و دلخواه P را دریافت کرده و با استفاده از یک زیرالگوریتم، حاصل جمع زیر را محاسبه کند.

$$\text{Sum} = 1! + 2! + \dots + p!$$

پاسخنامه

پاسخ آزمون نظری واحد کار ۱

- (ج-۱) (۲-الف) (۳-ج) (۴-ب) (۵-الف)

پاسخ آزمون نظری واحد کار ۲

- (ج-۱) (۲-ب) (۳-د) (۴-ب) (۵-ب)

فهرست منابع

- 1- Fundamental of data structures in C. (Horowitz, Sahni)
- 2- Algorithms and theory of Computation.(J.ATALLAH)

