

۳۷۲	واژه‌نامه
۳۷۲	خلاصه مطالب
۳۷۴	آزمون نظری
۳۷۶	آزمون عملی
	<b>واحد کار ۱۳: توانایی انجام یک پروژه عملی</b>
۴۰۱	آزمون پایانی
۴۰۶	پاسخنامه
۴۰۹	ضمایم
۴۴۲	فهرست منابع

## مقدمه

یکی از عمده‌ترین اهداف طراحی و تولید کامپیوترها انجام عملیات ذخیره‌سازی، بازیابی داده‌ها و اطلاعات و انجام انواع محاسبات به وسیله آن‌هاست. برای تحقق بخشیدن به این اهداف وجود دو جزء اصلی یعنی سخت‌افزار و نرم‌افزار الزامی است.

با پیدایش اولین کامپیوترها نیاز به وجود برنامه‌هایی که بتوان با به‌کارگیری آن‌ها کامپیوترها را مورد استفاده قرار داد، احساس شد و از آن‌جا که پردازش در کامپیوترها براساس مبنای باینری یا همان ۰ و ۱ است، اولین نرم‌افزارهایی که توسط متخصصین طراحی گردید به زبان ماشین (۰ و ۱) نوشته شد. این روش تولید نرم‌افزار کار دشوار و وقت‌گیری بود که محدودیت‌های زیادی را دربرداشت، بنابراین متخصصان علوم کامپیوتر تصمیم گرفتند تا نرم‌افزارهایی را تولید کنند که بتوانند با استفاده از آن‌ها هر نوع برنامه‌ای را با سرعت و دقت به زبان ماشین تبدیل نمایند. این امر منجر به تولید شاخه ویژه‌ای از نرم‌افزارها به نام زبان‌های برنامه‌نویسی شد. بدین ترتیب به موازات رشد و تکامل صنعت سخت‌افزار، زبان‌های برنامه‌نویسی کامپیوتر نیز خط سیر تکاملی خود را از زبان ماشین و اسمبلی به زبان‌های برنامه‌نویسی سطح بالا، ساخت یافته، شیء‌گرا و ویژوال، طی کرده و هر روزه زبان‌های برنامه‌نویسی کاربردی‌تری را در اختیار برنامه‌نویسان قرار دادند. در حال حاضر محدوده زبان‌های برنامه‌نویسی بسیار گسترده شده و با ظهور سیستم‌عامل‌های ویندوز و رایج‌شدن شبکه‌های کامپیوتری بخصوص اینترنت، این مسأله شدت بیشتری پیدا کرده است.

این پیمانه مهارتی با توجه به نیازهای آموزشی در رابطه با روش حل مسایل و زبان‌های برنامه‌نویسی سطح بالا و مدرن که قابلیت برنامه‌نویسی در محیط سیستم عامل ویندوز را نیز داشته باشند، تألیف شده است. زبان برنامه‌نویسی ویژوال بیسیک نسخه ۶ یکی از رایج‌ترین و کارآمدترین زبان‌های برنامه‌نویسی در دنیاست که از ویژگی‌های بالایی در برنامه‌نویسی حرفه‌ای برخوردار است. این پیمانه مهارتی شامل چهارده واحدکار است که به صورت خودآموز تهیه شده است. در واحدکار اول و دوم دانش‌آموزان با مفاهیم اساسی درباره الگوریتم و فلوچارت آشنا شده و چگونگی یافتن راه‌حل مناسب برای حل مسایل مختلف را به همراه مثال و تمرین‌های متعدد فرامی‌گیرند. از واحد کار سوم تا چهاردهم فراگیر می‌تواند مطالب را به صورت تئوری همراه با تصاویر مناسب مطالعه کرده و سپس مطالب مطالعه شده را با تمرین‌های مرتبط و به صورت عملی و مرحله‌ای انجام دهد. به علاوه سعی شده است که با ارایه آزمون‌های نظری و عملی در پایان هر واحدکار و یک آزمون نظری و عملی در پایان پیمانه مهارتی، مطالب مجدداً مرور شوند. برای بازدهی بالاتر در فراگیری این پیمانه مهارتی تسلط کافی در دروس پیش‌نیاز مانند مبنای و فناوری کامپیوتر، سیستم عامل مقدماتی و پیشرفته (ترجیحاً Windows XP) الزامی است.

## پیش‌آزمون

۱ - کدام کلید ترکیبی برای فعال کردن منوی Start به کار می‌رود؟

ب - Ctrl+Esc

الف - Alt+F4

د - Ctrl+C

ج - Alt+Esc

۲ - از کدام برنامه در ویندوز برای اضافه کردن یک زبان جدید به مجموعه زبان‌های

موجود استفاده می‌شود؟

ب - Regional and Language Options

الف - Keyboard

د - Regional Options

ج - Folder Options

۳ - کدام کلید ترکیبی برای حرکت کردن بین پنجره‌های باز مناسب است؟

ب - Alt+Tab

الف - Ctrl+V

د - Ctrl+X

ج - Alt+Esc



### توانایی حل مسایل و طراحی الگوریتم مناسب برای آنها

### هدف‌های رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- نحوه شناخت و بررسی مسایل 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 مسایل متفاوت طراحی نماید.

## 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، اساس روشی را در حل مسایل بنا نهاد 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 ارایه راه‌حل به دست آمده با توجه به قواعد علم الگوریتم را می‌آموزید.

## ۱-۱ شناخت مسایل و ارایه راه‌حل مناسب برای حل آن‌ها

انسان از آغاز آفرینش تاکنون همواره در مسیر زندگی خود با مشکلات و مسائل مختلفی روبه‌رو بوده و برای حل مشکلات خود، همواره راه‌حل‌های متفاوتی را تجربه کرده است. از مشکلاتی نظیر خوراک، پوشاک، محل زندگی تا حل مسائل علمی، فلسفی، ریاضی و نظایر آن‌ها؛ بنابراین با توجه به نوع مسایل و تجربیات و پیشرفت‌های علمی، سعی در ارائه راه‌حل‌های جدید کرده است. پیدا کردن راه‌حل برای یک مسأله به نوع آن بستگی دارد، بعضی از مسأله‌ها به سادگی قابل حل بوده اما بعضی دیگر هنوز هم به سادگی قابل حل نیستند.

برای حل هرگونه مسأله جدا از نوع آن می‌توان موارد زیر را در نظر گرفت:

۱- شناخت دقیق مسأله

۲- تجزیه و تحلیل مسأله

۳- طراحی راه‌حل

### ۱-۱-۱ شناخت مسأله

برای شناخت بهتر یک مسأله باید سه عامل مهم را در نظر بگیرید: مقادیر معلوم، خواسته‌های مسأله (مجهولات) و عملیات محاسباتی.

**مقادیر معلوم داده‌ها:** مقادیری که در اختیار مسأله قرار می‌گیرند و برای رسیدن به

هدف موردنظر در مسأله موردنیاز هستند.

**ارتباط بین داده‌ها و مجهول‌ها (محاسبات):** برای رسیدن به نتایج موردنظر معمولاً لازم است تا عملیاتی را روی مقادیر معلوم انجام دهید؛ بخش عمده‌ای از این عملیات 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 یک دایره به شعاع دلخواه را محاسبه کنیم. برای حل این مسأله با روش ارایه شده، ابتدا مقادیر معلوم را مورد توجه قرار می‌دهیم؛ همان‌طور که می‌دانید برای محاسبه محیط هر دایره باید شعاع آن را در اختیار داشته باشیم، بنابراین شعاع دایره (R) به عنوان تنها داده موردنیاز برای حل مسأله کافی است.

محاسباتی که برای رسیدن به محیط دایره لازم است، در واقع فرمول زیر خواهد بود:

$$P = 2 \times 3.14 \times R$$

با استفاده از این فرمول‌ها روابط بین داده‌های ورودی و نتایج موردنظر به خوبی تعیین می‌شود و بالاخره به عنوان مرحله آخر، خواسته مسأله (مجهول) که همان محیط (P) دایره است به دست می‌آید.



تمرین:

مقادیر معلوم، محاسبات و خواسته‌های مسأله (مجهولات) را در محاسبه محیط و مساحت یک مستطیل دلخواه مشخص کنید

## ۲-۱-۱ تجزیه و تحلیل مسأله

معمولاً جستجوی راه‌حل مناسب برای یک مسأله، به سادگی امکان‌پذیر نیست. گاهی اوقات راه‌حل مسأله به سادگی قابل تشخیص است، مانند محاسبه مجموع دو عدد یا محاسبه مساحت یک دایره اما در بعضی از مواقع به دلیل پیچیده بودن مسأله، لازم است

است مسأله اصلی به چند مسأله کوچکتر تقسیم شود که به آن‌ها زیرمسأله می‌گویند. سپس هر زیرمسأله مجدداً بررسی شود تا در صورت دشواری مجدداً به زیرمسأله‌های کوچکتر تقسیم شود و این عمل آنقدر ادامه یابد تا زیرمسأله‌ها به ساده‌ترین شکل حل شوند. در این حالت می‌توان هر زیرمسأله را جداگانه حل کرد و با کنار هم قرار دادن آن‌ها، به حل مسأله اصلی دسترسی پیدا کرد. به عنوان مثال اگر مسأله ساخت یک اتومبیل باشد پیدا کردن راه‌حل برای ساخت آن در مرحله اول، کار دشواری است اما اگر اتومبیل را به بخش‌های مختلف مثل موتور، جعبه‌دنده و اتاق و بدنه تقسیم کنیم و هر یک از این اجزا را مجدداً به کوچک‌ترین جزء، آن‌گاه با طراحی و ساخت اجزای کوچکتر می‌توان اجزای بزرگ‌تر را ایجاد کرد و در نهایت با ترکیب اجزای اصلی، طراحی و ساخت یک اتومبیل امکان‌پذیر خواهد شد.

### ۳-۱-۱ طراحی راه‌حل

پس از تحلیل مسأله، برای ارائه یک راه‌حل می‌توان به دو روش عمل کرد:

- ۱- استفاده از تجربیات و راه‌حل‌های موجود که در حل مسائل دیگر به کار گرفته شده‌اند.
- ۲- استفاده از روش‌های تفکر منطقی و الگوریتمی که حل مسأله براساس آن صورت گرفته و به صورت مرحله به مرحله انجام می‌شود. به عنوان مثال فرض کنید می‌خواهیم تعداد اعداد زوج بین اعداد ۱ تا ۲۰ را محاسبه کنیم، یک روش ساده این است که اعداد ۱ تا ۲۰ را نوشته و تعداد اعداد زوج را بشمارید، اما با استفاده از روش تفکر منطقی می‌توان به این نتیجه رسید که با توجه به این‌که اعداد زوج یکی در میان قرار می‌گیرند با تقسیم عدد ۲۰ بر ۲ به تعداد ۱۰ عدد دست می‌یابید که از روش اول هم به همین نتیجه خواهید رسید.

## ۲-۱ الگوریتم

### ۱-۲-۱ 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.



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 ارایه می‌کند.

در واقع 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. به طور 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 موردنیاز از خانه خارج می‌شوید، 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.

می‌توان این مراحل را به شکل خلاصه 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.

## ۲-۱- شرایط الگوریتم

پس از آشنایی با مفهوم الگوریتم، لازم است ویژگی های یک الگوریتم را بشناسید.  
**الف- استفاده از زبان ساده، دقیق و قابل فهم:** این ویژگی سبب می شود تا در انجام دستورالعمل ها همواره یک برداشت یکسان حاصل شود، در غیر این صورت برداشت های متفاوت سبب خواهد شد تا دستورالعمل ها نتایج متفاوتی را به وجود آورند. زبان الگوریتم نیز می تواند یکی از زبان های گفتاری و نوشتاری مانند فارسی، انگلیسی و ... باشد.

**ب- استفاده از جزییات کافی:** این ویژگی سبب می شود تا دستورالعمل ها به طور کامل اجرا شوند. وجود موارد نامشخص یا ارایه دستورالعمل ها به صورت کلی و مبهم سبب مخدوش شدن نتایج خواهد شد.

**ج- شروع و پایان الگوریتم:** در یک الگوریتم باید شروع دستورالعمل ها مشخص باشد. هر الگوریتم یک نقطه شروع دارد که به عنوان اولین دستورالعمل از آن استفاده می شود، به علاوه پایان الگوریتم نیز باید تعیین شود. یک الگوریتم می تواند بیش از یک نقطه پایان داشته باشد.

**د- ترتیب انجام دستورالعمل ها:** یکی از ویژگی های مهم یک الگوریتم ترتیب اجرای دستورالعمل ها است؛ اگر این کار به درستی انجام نشود، پیش بینی نتیجه کار مشخص نخواهد بود. در یک الگوریتم ترتیب انجام عملیات با استفاده از شماره گذاری دستورالعمل ها از بالا به پایین انجام می شود که البته در صورت نیاز می توان ترتیب اجرای دستورالعمل ها را نیز تغییر داد. در مباحث بعد به این مسأله خواهیم پرداخت.



**نکته:** اگر در حل مسائل سه عامل اصلی را به دقت مشخص کنید، طراحی یک الگوریتم کار زیاد دشواری نخواهد بود. اگر مقادیر معلوم، خواسته های مسأله، فرمول ها و روابط ریاضی و منطقی بین آن ها را به درستی تعیین کنید، نوشتن الگوریتم های مختلف آسان تر خواهد شد.

**هـ - جامع بودن:** الگوریتم باید به شکلی طراحی شود که با توجه به صورت مسأله و مفروضات آن در تمام حالت ها، نتایج مناسب و صحیحی را ارایه کرده و در حالت های خاص یا داده های ورودی متفاوت، نتایج درستی را ایجاد کند.



**نکته:** بهتر است در یک الگوریتم 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

الگوریتم مجموعه‌ای از دستورالعمل‌هاست. دستورالعمل‌ها انواع مختلفی دارند که عبارتند از:

- 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 مانند «نمایش بده یا چاپ کن» به کار می‌روند.

#### ج- دستورالعمل‌های محاسباتی

این دستورالعمل‌ها در واقع نحوه ارایه 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 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: برخی از عملیاتی که در الگوریتم انجام می‌گیرد، عملیات ریاضی و محاسباتی است که برای انجام این گونه دستورات لازم است از عملگرهای ریاضی استفاده شود.

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

جدول ۱-۱

اولویت	مثال	عنوان	عملگر
۱	$4 + (2 \times 3) = 10$	پرانتز	( )
۲	$2 \times 3 = 6$	ضرب	*
۲	$12 / 6 = 2$	تقسیم	/
۳	$17 \div 3 = 5$	خارج قسمت تقسیم صحیح	\
۴	$14 \bmod 5 = 4$	باقی‌مانده تقسیم صحیح	Mod
۵	$35 + 14 = 49$	جمع	+
۵	$10 - 7 = 3$	تفریق	-

عملگرهای ارایه 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 استفاده از چند عملگر ریاضی، اولویت آن‌ها مطابق جدول ۱-۱ از بالا به پایین در نظر گرفته می‌شود، به عنوان مثال عبارت  $۱۰/۵ \times ۳ + ۲$  را در نظر بگیرید؛ در این عبارت 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  $۱/۵$  با عدد ۲ جمع و در نتیجه عدد  $۳/۵$  به دست می‌آید.

توجه کنید که عملگرهای ضرب و تقسیم نسبت به هم، جمع و تفریق نیز نسبت به هم دارای اولویت یکسانی هستند و اولویت آن‌ها نسبت به هم از چپ به راست است. به عنوان مثال عبارت  $۲ \times ۶ - ۴۰ + ۱۰$  را در نظر بگیرید؛ در این عبارت ابتدا عدد ۶ در ۲ ضرب شده، سپس عبارت  $۴۰ + ۱۰$  محاسبه می‌شود که نتیجه آن عدد ۵۰ خواهد بود و بعد عمل تفریق انجام می‌شود و حاصل تفریق  $۱۲ - ۵۰$  یعنی ۳۸ به دست می‌آید.

البته اولویت عملگرها را می‌توان با پرانتز تغییر داد به عنوان مثال اگر عبارت قبل را با پرانتز به شکل  $۲ \times (۱۰ + ۴۰ - ۶)$  تغییر دهید ابتدا عبارت داخل پرانتز محاسبه می‌شود سپس نتیجه در عدد ۲ ضرب خواهد شد.

### تمرین:



ترتیب انجام عملیات و حاصل عبارات زیر را تعیین کنید:

$$(۱۴ - ۳ \times ۲) / (۵ + ۲)$$

$$((۲۷ + ۴) \times ۱۲ - ۷) + ۸ / ۲$$




مثال ۱: فرض کنید می‌خواهید مسأله محاسبه

محیط یک مستطیل با طول  $L$  و عرض  $W$  دلخواه را به صورت یک الگوریتم بنویسید، قبلاً این مسأله را بررسی کرده 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	
۱- شروع الگوریتم	شروع الگوریتم
۲- $L$ را دریافت کن	ورود داده‌ها
۳- $W$ را دریافت کن	ورود داده‌ها
۳- $P \leftarrow 2 \times (L + W)$	محاسبه محیط
۴- $P$ را نمایش بده	نمایش خروجی
۵- پایان	پایان الگوریتم

### تمرین:

الگوریتمی بنویسید که محیط یک دایره با شعاع دلخواه R را محاسبه کرده و نمایش دهد.

مثال ۲: الگوریتمی بنویسید تا حاصل تقسیم یک عدد دلخواه را بر عدد دلخواه دیگری محاسبه کرده و نمایش دهد. 

در این مثال ورودی‌ها شامل دو عدد هستند و محاسبات لازم برای رسیدن به نتیجه، عمل ریاضی تقسیم می‌باشد و خواسته مسأله نیز نتیجه تقسیم دو عدد بر یکدیگر است، بنابراین الگوریتم مربوطه بدین صورت خواهد بود.

توضیح	
۱- شروع	شروع
۲- X را دریافت کن	ورود داده‌ها
۳- Y را دریافت کن	ورود داده‌ها
۳- $Z \leftarrow X/Y$	محاسبه حاصل تقسیم
۴- Z را نمایش بده	نمایش خروجی
۵- پایان	پایان الگوریتم

در این الگوریتم پس از شروع الگوریتم در مراحل ۲ و ۳ دو عدد از ورودی دریافت شده و در متغیرهای X و Y ذخیره می‌شوند. در مرحله ۴ دو متغیر بر هم تقسیم شده و نتیجه تقسیم در متغیر Z ذخیره می‌شود. در مرحله ۵ حاصل تقسیم روی صفحه نمایش داده می‌شود و در مرحله ۶ الگوریتم پایان می‌یابد.

### تمرین:

الگوریتمی بنویسید تا میانگین سه عدد دلخواه را محاسبه کرده و نمایش دهد.  
الگوریتمی بنویسید که باقیمانده تقسیم دو عدد دلخواه را بر هم محاسبه کرده و نمایش دهد.

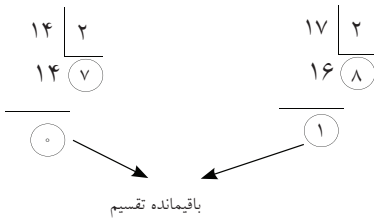
## د - دستورالعمل‌های شرطی

گاهی اوقات لازم است با مقایسه مقادیر داده‌های ورودی، اطلاعات خروجی، متغیرها و ... 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 می‌توانید از دستورات شرطی استفاده کنید.



مثال ۳: الگوریتمی بنویسید که زوج و فرد بودن هر عدد دلخواه را مشخص کند.

در این مسأله داده ورودی شامل یک عدد است و تشخیص زوج یا فرد بودن عدد وارد شده خواسته (مجهول) مسأله است. اگر باقیمانده تقسیم یک عدد بر عدد ۲ صفر شود عدد زوج است و اگر باقیمانده یک باشد عدد موردنظر فرد می‌باشد.



با استفاده از عملگر Mod می‌توان تقسیم‌های فوق را به صورت زیر نوشت:

$$14 \text{ MOD } 2 \rightarrow 0$$

$$17 \text{ MOD } 2 \rightarrow 1$$

در این مرحله با این‌که نکات لازم برای حل مسأله به دست آمده است اما یک مشکل هنوز وجود دارد که یک عدد در هر لحظه نمی‌تواند هم زوج و هم فرد باشد، بنابراین مجبور خواهید بود در زمانی که عدد زوج است یک خروجی و وقتی عدد فرد است خروجی دیگری نمایش داده شود. در چنین حالت‌هایی لازم است از دستورات شرطی استفاده کنید. نحوه استفاده از یک دستور شرطی به صورت زیر است:

شکل کلی دستورالعمل شرطی به یکی از صورت‌های زیر است.

اگر شرط (ها) آن‌گاه دستور(ات)

اگر شرط (ها) آن‌گاه دستور(ات) در غیر این صورت دستور(ات)

در حالت اول، ابتدا شرط یا شرط‌های ارایه شده بررسی می‌شوند و در صورتی که نتیجه بررسی درست باشد، دستور یا دستورات پس از «آن‌گاه» اجرا می‌شوند، در غیراین صورت (نادرست بودن شرط بررسی شده) دستورالعملی که پس از دستورالعمل شرطی قرار گرفته، اجرا خواهد شد بدون آن‌که دستور یا دستورات پس از «آن‌گاه» 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، دستورالعمل شرط مانند حالت اول رفتار می‌کند اما اگر نتیجه ارزیابی شرط یا شرط‌ها نادرست باشد بدون آن‌که دستورات بخش «آن‌گاه» اجرا شوند، دستورات موجود در بخش «در غیر این صورت» اجرا خواهند شد سپس دستورالعملی که پس از دستور شرطی قرار دارد، اجرا می‌شود.

جدول ۲ - ۱

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

بنابراین الگوریتم موردنظر بدین صورت خواهد بود:

توضیح	
۱- شروع	شروع الگوریتم
۲ - $X$ را دریافت کن	ورود داده
۳ - $2 \leftarrow X \text{ Mod } R$	مقدار باقیمانده تقسیم را در $R$ ذخیره می‌کند.
۴ - اگر $R = 0$ آن‌گاه «عدد زوج است.» را نمایش بده.	در صورتی که $R$ برابر صفر باشد عبارت «عدد زوج است.» را نمایش می‌دهد.
۵ - اگر $R = 1$ آن‌گاه «عدد فرد است.» را نمایش بده.	در صورتی که $R$ برابر یک باشد عبارت «عدد فرد است.» را نمایش می‌دهد.
۶- پایان	پایان الگوریتم



پس از شروع الگوریتم، در مرحله ۲ یک عدد از ورودی دریافت شده و در متغیر  $X$  ذخیره می‌شود سپس در مرحله ۳ باقیمانده تقسیم  $X$  بر ۲ محاسبه شده و در متغیر  $R$  ذخیره می‌شود. در مرحله ۴ با استفاده از یک دستور شرطی مقدار متغیر  $R$  با مقدار صفر مقایسه می‌شود در صورتی که مقدار متغیر نیز صفر باشد نتیجه بررسی شرط درست (True) و در نتیجه عدد زوج خواهد بود و عبارت «عدد زوج است.» نمایش داده می‌شود اما اگر مقدار متغیر  $R$  صفر نباشد نتیجه بررسی شرط نادرست (False) بوده و دستور بعد از آن‌گاه اجرا نمی‌شود. در مرحله ۵ مجدداً مقدار  $R$  بررسی می‌شود و در صورتی که مقدار آن برابر با یک باشد «عبارت عدد فرد است.» نمایش داده خواهد شد و در صورتی که مقدار آن برابر با یک نباشد عبارت مزبور نمایش داده نمی‌شود.



تمرین:

الگوریتمی بنویسید که بخش‌پذیری یک عدد را بر عدد دیگر تعیین کند

با توجه به مطالب گفته شده و این نکته که یک عدد نمی‌تواند هم زوج و هم فرد باشد الگوریتم مثال قبل را به صورت زیر می‌توان خلاصه‌تر نمود و به جای استفاده از دو دستورالعمل شرطی مسأله را با یک دستورالعمل شرطی حل کرد.

توضیح	
شروع الگوریتم	۱- شروع
ورود داده	۲ - $X$ را دریافت کن
مقدار باقیمانده تقسیم را در $R$ ذخیره می‌کند.	۳ - $2 \text{ MoD } X \leftarrow R$
در صورتی که مقدار $R$ برابر با صفر باشد عبارت «عدد زوج است.» را نمایش می‌دهد، در غیر این صورت عبارت «عدد فرد است» را نمایش می‌دهد.	۴ - اگر $R = 0$ آن‌گاه «عدد زوج است.» را نمایش بده، در غیر این صورت «عدد فرد است.» را نمایش بده
پایان الگوریتم	۵- پایان

در این الگوریتم جدید از یک دستورالعمل شرطی استفاده شده است اما در نتیجه اجرای الگوریتم هیچ تغییری به وجود نمی‌آید، در این حالت اگر مقدار باقیمانده یعنی متغیر  $R$  برابر با صفر باشد دستوری که در بخش «آن‌گاه» قرار دارد اجرا می‌شود اما اگر مقدار متغیر  $R$  برابر با صفر نباشد دستور بخش «در غیر این صورت» اجرا خواهد شد.

## تمرین:



الگوریتم بخش پذیری دو عدد بر یکدیگر را با یک دستورالعمل شرطی بنویسید.



مثال ۴: 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 می گیریم، برای پیدا کردن بزرگترین مقدار در بین این سه عدد می توانیم از روش مقایسه استفاده کنیم؛ به این منظور ابتدا فرض می کنیم مقدار ذخیره شده در متغیر  $A$  بزرگترین مقدار است و مقدار آن را در متغیر دیگری به نام  $MAX$  ذخیره می کنیم، به این دلیل که متغیرهای دیگر یعنی  $B$  و  $C$  را با مقایسه با این مقدار (متغیر  $MAX$ ) بررسی کرده و بزرگتر بودن یا نبودن آنها را از مقدار اول مشخص می کنیم یعنی متغیر  $B$  را با  $MAX$  مقایسه کرده و در صورت بزرگتر بودن مقدار  $B$  از  $MAX$ ، متغیر  $B$  را در  $MAX$  ذخیره می کنیم، 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 می شود.	۳ - $MAX \leftarrow A$
در صورتی که مقدار $B$ از $MAX$ بزرگتر باشد مقدار $B$ را در $MAX$ ذخیره می کند.	۴- اگر $B > MAX$ آن گاه $MAX \leftarrow B$
در صورتی که مقدار $C$ از $MAX$ بزرگتر باشد مقدار $C$ را در $MAX$ ذخیره می کند.	۵ - اگر $C > MAX$ آن گاه $MAX \leftarrow C$
نمایش خروجی	۶ - $MAX$ را نمایش بده
پایان الگوریتم	۷ - پایان

## 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.



**نکته** در صورت نیاز، می‌توان یک دستورالعمل شرطی را در بخش دستورات بعد از «آن‌گاه» یا «در غیر این صورت» دستورالعمل شرطی دیگری قرار داد؛ در این حالت به مجموعه دستورالعمل‌های شرطی، دستورالعمل‌های شرطی تودرتو نیز می‌گویند.

## عملگرهای منطقی

گاهی اوقات ممکن است لازم باشد در دستورالعمل‌های شرطی که در حل یک الگوریتم به کار می‌روند بیش از یک شرط مورد بررسی قرار گیرند و با توجه به نتیجه بررسی تمامی شرط‌های ذکر شده سایر دستورالعمل‌ها اجرا شوند. در چنین حالت‌هایی می‌توان از عملگرهای منطقی برای ترکیب دو یا چند شرط استفاده کرد.



مثال ۵: الگوریتمی بنویسید که مقدار مصرف برق یک مشترک دلخواه را دریافت کرده و براساس جدول ۱-۳، رتبه میزان مصرف وی را مشخص کند.

جدول ۱-۳

ردیف	میزان مصرف (کیلووات)	رتبه
۱	۰ تا ۵۰	کم مصرف
۲	۵۰ تا ۲۰۰	عادی
۳	۲۰۰ به بالا	پرمصرف

برای طراحی الگوریتمی که بتواند رتبه مصرف برق یک مشترک را مشخص 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 ردیف‌های ۱ و ۲ در جدول ۳-۱ از یک دستورالعمل شرطی در بخش «آن‌گاه» دستورالعمل شرطی دیگری استفاده شود تا هر دو شرط موردنظر در این ردیف‌ها مورد ارزیابی قرار گیرند. البته برای ردیف ۳ می‌توان از یک دستورالعمل شرطی به تنهایی استفاده کرد. با توجه به این توضیحات می‌توانید الگوریتم زیر را بنویسید:

در این الگوریتم kw به عنوان میزان مصرف برق در نظر گرفته شده است.

۱- شروع

۲- kw را دریافت کن.

۳- اگر  $kw \geq 0$  آن‌گاه اگر  $kw < 50$  آن‌گاه «مشترک کم مصرف است.» را نمایش بده.

۴- اگر  $kw \geq 50$  آن‌گاه اگر  $kw < 200$  آن‌گاه «مشترک عادی است.» را نمایش بده.

۵- اگر  $kw \geq 200$  آن‌گاه «مشترک پرمصرف است.» را نمایش بده.

۶- پایان

به طور کلی سه نوع عملگر منطقی که در الگوریتم مورد استفاده قرار می‌گیرند در

جدول ۴-۱ ارائه شده‌اند.

جدول ۴-۱

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 درست (True) خواهد بود که نتیجه تمام شرط‌ها و مقایسه‌ها True باشند؛ به عنوان مثال شرط‌های  $X > 0$  و  $Y < 5$  را در نظر بگیرید. اگر بخواهیم این دو شرط را با یکدیگر ترکیب کنیم نتیجه ترکیب آن‌ها زمانی درست (True) خواهد بود که نتیجه بررسی هر دو شرط درست (True) باشد. به عنوان نمونه اگر مقدار X برابر ۲ و Y برابر ۱ باشد هر دو شرط درست (True) بوده و نتیجه ترکیب آن‌ها نیز درست (True) خواهد بود.

اما اگر نتیجه بررسی شرط  $X > 0$  یا  $Y < 5$  یا هر دو شرط نادرست (False) باشد، نتیجه ترکیب آن‌ها نیز نادرست (False) خواهد بود. به عنوان نمونه اگر مقدار  $X$  برابر با ۵- و مقدار  $Y$  برابر با ۱ باشد شرط  $X > 0$  نادرست (False) و شرط  $Y < 5$  درست (True) خواهد بود و در نتیجه ترکیب آن‌ها نیز نادرست (False) می‌شود. بنابراین جدول درستی عملگر منطقی AND مطابق جدول ۱-۵ است. در صورتی که دو یا چند شرط را با عملگر منطقی OR با هم ترکیب کنید، نتیجه شرط زمانی نادرست (False) خواهد بود که نتیجه تمام شرط‌ها نادرست (False) باشد و در سایر حالات، نتیجه درست (True) خواهد بود؛ بنابراین جدول درستی عملگر منطقی OR مطابق جدول ۱-۶ است.

جدول ۱-۵

$X > 0$	$Y < 5$	$X > 0 \text{ AND } Y < 5$
T	F	F
F	T	F
F	F	F
T	T	T

جدول ۱-۶

$X > 0$	$Y < 5$	$X > 0 \text{ OR } Y < 5$
T	F	T
F	T	T
F	F	F
T	T	T

اگر از عملگر NOT روی یک شرط استفاده شود نتیجه درستی آنرا معکوس می‌کند یعنی اگر نتیجه شرط درست (True) باشد آنرا به نادرست (False) و اگر نتیجه شرط نادرست (False) باشد آنرا به درست (True) تبدیل می‌کند. جدول درستی عملگر منطقی NOT مطابق جدول ۱-۷ می‌باشد.

جدول ۱-۷

$X > 0$	$\text{NOT } X > 0$
T	F
F	T

**نکته** در جدول‌های ۱-۵، ۱-۶ و ۱-۷، T بیانگر درست (True) و F بیانگر نادرست (False)

می‌باشد.



### تمرین:



الگوریتمی بنویسید تا معدل یک دانش آموز را دریافت کرده و رتبه وی را براساس شرایطی که در ادامه می آید، تعیین کند:

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 باشد، رتبه خوب برای وی اعلام شود.

ج- در صورتی که میانگین نمرات بین ۱۲ تا ۱۶ باشد، رتبه متوسط برای وی اعلام شود.

د- در صورتی که میانگین نمرات کمتر از ۱۲ باشد، رتبه ضعیف برای وی اعلام شود.

### هـ - دستورالعملهای تکرار (حلقه‌ها)

استفاده از دستورالعمل‌هایی که تاکنون فراگرفته‌اید در حل بعضی از مسایل کافی نیست و لازم است تا برخی از دستورالعمل‌ها را به دفعات تکرار نمایید؛ در این موارد از دستورالعمل تکرار یا همان حلقه‌ها استفاده کنید.

مثال ۶: الگوریتمی بنویسید که مجموع اعداد ۱ تا ۱۰ را محاسبه نموده و نمایش دهد.



اگر برای حل این مسأله بخواهید از دستورالعمل‌هایی که تاکنون ارائه شده‌اند استفاده کنید در این حالت الگوریتمی به صورت زیر به دست می آید.

۱ - شروع

۲ -  $SUM \leftarrow 0$

۳ -  $SUM = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$

۴ - SUM را نمایش بده

۵ - پایان

با این حال مشاهده می کنید که الگوریتم شکل نامناسبی دارد و اگر داده‌ها زیادتر شوند استفاده از این روش عملاً غیرممکن است. برای حل این مشکل می توان دستورالعمل تکرار را به کار برد.

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 داخل حلقه تکرار می‌شوند. این دستورالعمل‌ها با توجه به نیاز مسأله انتخاب می‌شوند.

بنابراین می‌توان الگوریتم مثال ۸ را به صورت زیر را ارائه نمود:

توضیح	۱- شروع
تنظیم مقدار اولیه برای شمارنده حلقه	۲- $N \leftarrow 1$
	۳- $SUM \leftarrow 0$
	۴- $SUM \leftarrow SUM + N$
محاسبه مجموع اعداد	۵- $N \leftarrow N + 1$
	۶- اگر $N \leftarrow 10$ آن‌گاه به مرحله ۴ برو
	۷- $SUM$ را نمایش بده
<div style="display: flex; align-items: center;"> <div style="font-size: 3em; margin-right: 10px;">{</div> <div>                         دستورات حلقه                         <div style="margin-top: 10px;">                             افزایش شمارنده حلقه                              محاسبه مجموع اعداد                              شرط حلقه                         </div> </div> </div>	۸- پایان

در این الگوریتم پس از شروع، ابتدا عدد ۱ در متغیر  $N$  ذخیره می‌شود که به عنوان شمارنده حلقه از آن استفاده می‌شود. در مرحله بعد مقدار صفر در متغیر  $SUM$  ذخیره می‌شود که برای محاسبه مجموع اعداد از آن استفاده می‌شود. در مرحله ۴ مجموع اعداد محاسبه می‌شود.

در مرحله بعد مقدار شمارنده حلقه  $N$  یک واحد افزایش می‌یابد، در مرحله ۶ یک

جدول ۸-۱

N	SUM	خروجی
<del>۱</del>	<del>۱</del>	۵۵
<del>۲</del>	<del>۳</del>	
<del>۳</del>	<del>۶</del>	
<del>۴</del>	<del>۱۰</del>	
<del>۵</del>	۱۵	
<del>۶</del>	۲۱	
۷	۲۸	
<del>۸</del>	<del>۳۶</del>	
<del>۹</del>	<del>۴۵</del>	
۱۰	۵۵	
۱۱	۵۵	

دستورالعمل شرطی مقدار شمارنده را بررسی می کند تا تعداد دفعات تکرار دستورالعمل ها از مقدار ۱۰ بیشتر نشود. در این حالت اگر مقدار شمارنده کوچک تر یا مساوی ۱۰ باشد اجرای الگوریتم به مرحله ۴ منتقل می شود. دستورات مراحل ۴، ۵ و ۶ آن قدر تکرار می شوند تا مقدار شمارنده از مقدار ۱۰ بیشتر شود و در نتیجه اجرای حلقه ایجاد شده خاتمه می یابد و اجرای الگوریتم با اجرای مرحله ۷ ادامه یافته و با نمایش میانگین اعداد در مرحله ۸ خاتمه می یابد. اگر الگوریتم را اجرا و آزمایش کنید جدول بررسی عملکرد آن به صورت جدول ۸-۱ خواهد بود.

مثال ۷: الگوریتمی بنویسید که اعداد طبیعی کوچک تر از ۵۰ را نمایش دهد.



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

مثال ۸: الگوریتمی بنویسید که اعداد زوج کوچک تر 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 می یابد.





توضیح	
	۱- شروع
	۲- $N$ را دریافت کن
خاتمه الگوریتم در صورتی که داده ورودی عدد یک باشد.	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  $100$  عدد  $3$  را نمایش دهد.



مثال ۹: 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-  $K \leftarrow M+1$
- ۵-  $SUM \leftarrow SUM+K$
- ۶-  $NO \leftarrow NO+1$
- ۷-  $K \leftarrow K+1$
- 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 NO برای شمارش تعداد اعداد بین M و N استفاده می شود. متغیر K برای ایجاد اعداد طبیعی بین M و N به کار می رود. در صورت اجرای این الگوریتم به ازای  $M=3$  و  $N=9$ ، جدول بررسی عملکرد به صورت زیر خواهد بود:

جدول ۹-۱

M	N	K	SUM	NO	خروجی
۳	<del>۹</del>	<del>۴</del>	<del>۶</del>	<del>۶</del>	
	<del>۴</del>	<del>۵</del>	<del>۴</del>	<del>۴</del>	
	<del>۳</del>	<del>۶</del>	<del>۹</del>	<del>۴</del>	
	<del>۲</del>	<del>۷</del>	<del>۱۵</del>	<del>۳</del>	
	<del>۱</del>	<del>۸</del>	<del>۲۴</del>	<del>۴</del>	
۳	۹	۳۰	۵	۳۰	۵

تمرین:



الگوریتم قبل را به گونه ای تنظیم کنید که بدون در نظر گرفتن فرض  $M < N$ ، توانایی ارائه پاسخ صحیح را داشته باشد، به عبارت دیگر در صورتیکه کاربر عدد بزرگتر را برای M و عدد کوچکتر را برای N وارد کند، پاسخ الگوریتم درست باشد.



**مثال ۱۰:** الگوریتمی بنویسید که مقسوم علیه‌های عدد طبیعی و دلخواه  $M$  را

محاسبه کند. به این منظور باید از یک حلقه استفاده کنید و متغیر  $M$  را هر بار بر یک متغیر  $K$  که به عنوان مقسوم علیه 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 مقسوم علیه متغیر  $M$  باشد، بنابراین متغیر  $K$  نمایش داده می‌شود. برای محاسبه باقیمانده تقسیم صحیح نیز می‌توانید از عملگر  $\text{Mod}$  استفاده کنید.

اکنون الگوریتم موردنظر به صورت زیر خواهد بود:

۱ - شروع

۲ -  $M$  را دریافت کن

۳ -  $K \leftarrow 1$

۴ - اگر  $M \text{ Mod } K = 0$  آن‌گاه  $K$  را نمایش بده

۵ -  $K \leftarrow K + 1$

۶ - اگر  $K \leq M$  آن‌گاه 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 قبل و محاسبه مجموع ارقام یک عدد، ابتدا باید ارقام عدد را یکی یکی از آن جدا کنید؛ به این منظور می‌توانید از عملگر  $\text{Mod}$  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:

$$\begin{array}{r} 17 \overline{) 5} \\ 15 \quad \text{← خارج قسمت تقسیم صحیح} \\ \hline 2 \quad \text{← باقیمانده تقسیم صحیح} \end{array}$$

$$17 \div 5 = 3$$

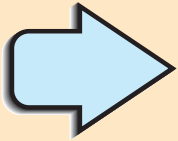
با توجه به مطالب ارایه شده، اگر بخواهیم ارقام یک عدد را از آن جدا کنیم روش ریاضی این عملیات به صورت بعد خواهد بود:

به عنوان مثال اگر عدد مورد نظر ۴۲۵۷ باشد:

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

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

- ۱ - شروع
- ۲ -  $M$  را دریافت کن
- $SUM \leftarrow 0$  - ۳
- $DIGIT \leftarrow M \bmod 10$  - ۴
- $SUM \leftarrow SUM + DIGIT$  - ۵
- $M \leftarrow M \div 10$  - ۶
- ۷ - اگر  $M > 0$  آن‌گاه برو به مرحله ۴
- ۸ -  $SUM$  را نمایش بده
- ۹ - پایان



## Learn in English

### Algorithm

To make a computer do anything, you have to write a computer program. To write a computer program, you have to tell the computer, step by step, exactly what you want it to do. The computer then «executes» the program, following each step mechanically, to accomplish the end goal.

When you are telling the computer *what* to do, you also get to choose how it's going to do it. That's where computer algorithms come in. The algorithm is the basic technique used to get the job done. Let's follow an example to help get an understanding of the algorithm concept.

Let's say that you have a friend arriving at the airport, and your friend needs to get from the airport to your house. Here are four different algorithms that you might give your friend for getting to your home:

#### \* **The taxi algorithm**

1 - Go to the taxi stand

Get in a taxi

2 - Give the driver my address

#### \* **The call-me algorithm:**

1 - When your plane arrives, call my cell phone.

2 - Meet me outside baggage claim.

#### \* **The rent-a-car algorithm:**

1 - Take the shuttle to the rental car place.

2 - Rent a car.

3 - Follow the directions to get to my house.

#### \* **The bus algorithm:**

1 - Outside baggage claim.

2 - Transfer to bus on Main Street.

3 - Get off on 20 street.

#### 4 - Walk two blocks north to my house.

All four of these algorithms accomplish exactly the same goal, but each algorithm does it in completely different way. Each algorithm also has a different cost and a different travel time. Taking a taxi, for example, is probably the fastest way, but also the most expensive. Taking the bus is definitely less expensive, but a whole lot slower. You choose the algorithm based on the circumstances.

## واژه‌نامه

Circumstance	وضعیت
Driver	راننده
Execute	اجرا کردن
Expensive	گران
Fast	سریع
Program	برنامه
Rent	اجاره

## خلاصه مطالب

- برای دستیابی به روش حل مسایل، شناسایی سه پارامتر مقادیر معلوم، محاسبات و خواسته‌های مسأله (مجهولات) کار را آسانتر می‌کند.
- الگوریتم مجموعه‌ای از دستورالعمل‌هاست که به صورت مرحله به مرحله اجرا می‌شوند و هدف مشخصی را دنبال می‌کنند و دارای شروع و خاتمه می‌باشند.
- 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.

## 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  $I \leq 100$  آن گاه برو به مرحله ۴

۷ -  $SUM$  را نمایش بده

۸ - پایان

الف - اعداد ۱ تا ۹۹ را نمایش می دهد

ب - مجموع اعداد ۰ تا ۹۹ را نمایش می دهد.

ج - مجموع اعداد ۱ تا ۱۰۰ را نمایش می دهد.

د - مجموع اعداد ۱ تا ۹۹ را نمایش می دهد.

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  $I < 100$  آن گاه برو به مرحله ۵ در غیر این صورت برو به مرحله ۸

۸ - پایان

ب - ۲ و ۴ و ۶ و ۸

الف - ۲ و ۴ و ۴ و ۴ و ۴

د - ۲ و ۴ و ۴ و ۴ و ۴

ج - ۲ و ۴ و ۶ و ۸ و ۱۰

۳ - نتیجه اجرای الگوریتم زیر چیست؟

۱ - شروع

۲ -  $I \leftarrow 1$

۳ - اگر  $I \bmod 2 = 0$  آن گاه  $I$  را نمایش بده.



$$I \leftarrow I + 1 - 4$$

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 صورتیکه  $X = 5$ ،  $Y = -10$  و  $Z = 27$  باشد، نتیجه عبارت زیر چیست؟

( $X > 0$ ) یا ( $Y < 20$ ) و ( $Z > X$ )

الف- درست      ب- نادرست      ج- ۱-      د- ۰

۶ - حاصل عبارت زیر کدام است؟

$$5 + 10 - 20 / 4 \times 3$$

الف- ۰      ب-  $3/75$       ج- ۳      د- ۱۳

۷ - حاصل عبارت زیر چیست؟

$$17 \text{ Mod } 3$$

الف- ۲      ب- ۳      ج- ۴      د- ۵

۸ - حاصل عبارت زیر چیست؟

$$25 \div 4$$

الف- ۱      ب- ۲      ج- ۵      د- ۶

۹ - در کدام مرحله از مراحل حل یک مسأله ارتباط بین داده‌ها و مجهول‌ها مشخص

می‌شود؟

الف- شناخت مسأله      ب- تجزیه مسأله

ج- طراحی راه‌حل      د- تحلیل مسأله

۱۰ - نتیجه اجرای الگوریتم زیر چیست؟

۱ - شروع

۲ -  $K \leftarrow 0$  و  $SUM \leftarrow 0$

۳- N را دریافت کن

۴- اگر  $N \text{ Mod } K = 0$  آن گاه  $SUM \leftarrow SUM + K$

۵-  $K \leftarrow K + 1$

۶- اگر  $K \leq N$  آن گاه برو به مرحله ۴

۷- SUM را نمایش بده

۸- پایان

الف- مجموع باقیمانده تقسیم N بر K      ب- مجموع مقسوم علیه های عدد N

ج- مجموع مقسوم علیه های عدد K      د- مجموع مقسوم علیه های عدد K بر N

۱۱- The algorithm is the basic technique used to write a ..... for running in the computer.

a- commands

b- statments

c- program

d- step by step

۱۲- الگوریتم را تعریف کنید و برای آن یک مثال بنویسید.

۱۳- مراحل حل مسأله را نام ببرید.

۱۴- انواع دستورالعمل ها را در الگوریتم نام ببرید.

۱۵- شرایط اصلی یک الگوریتم خوب را توضیح دهید.

۱۶- کاربرد عملگرهای منطقی و انواع آنها را توضیح دهید.

۱۷- دستورالعمل های شرطی و کاربرد آنها را توضیح دهید.

۱۸- دستورالعمل های تکرار و کاربرد آنها را توضیح دهید.

۱۹- عوامل مؤثر در شناخت و حل مسائل را توضیح دهید.

۲۰- منظور از جامع بودن الگوریتم چیست؟

۲۱- عوامل اصلی در شناخت دقیق یک مسأله را توضیح دهید.

## آزمون عملی

- ۱ - الگوریتمی بنویسید که مضارب کوچک‌تر از ۱۰۰۰ عدد ۵ را محاسبه کرده و نمایش دهد.
- ۲ - الگوریتمی بنویسید که دما را برحسب فارنهایت دریافت کرده و معادل آن را به سانتی‌گراد نمایش دهد (با توجه به فرمول  $C = \frac{5}{9}(F - 32)$ ).
- ۳ - عدد طبیعی و دلخواه N را دریافت کرده و N! را محاسبه کرده و نمایش دهد.
- ۴ - الگوریتمی بنویسید که حقوق یک کارمند را دریافت کرده و میزان مالیات حقوق وی را مطابق جدول زیر پس از محاسبه نمایش دهد.

مالیات	حقوق
صفر	کمتر از ۴/۰۰۰/۰۰۰ ریال
۲ درصد حقوق	از ۴/۰۰۰/۰۰۰ ریال تا ۵/۵۰۰/۰۰۰ ریال
۳ درصد حقوق	از ۵/۵۰۰/۰۰۰ ریال تا ۷/۰۰۰/۰۰۰ ریال
۵ درصد حقوق	از ۷/۰۰۰/۰۰۰ ریال به بالا

- ۵ - الگوریتمی بنویسید که نمرات ۱۰ درس یک دانش‌آموز را دریافت کرده و معدل و میانگین نمرات وی را نمایش دهد.
- ۶ - الگوریتمی بنویسید که عدد طبیعی و دلخواه N را دریافت کرده و ارقام آن را معکوس کند (مثلاً عدد ۲۴۸۵ به ۵۸۴۲ تبدیل شود).
- ۷ - الگوریتمی بنویسید که مقدار زمان را براساس ثانیه دریافت کرده و مقدار ساعت‌ها، دقیقه‌ها و ثانیه‌های آن را محاسبه کرده و نمایش دهد.
- ۸ - الگوریتمی بنویسید که عدد طبیعی و دلخواه M را دریافت کرده و اعداد زوج کوچک‌تر از آن را نمایش دهد.
- ۹ - الگوریتمی بنویسید که دو عدد طبیعی دلخواه را دریافت کرده و بزرگ‌ترین مقسوم‌علیه مشترک آن‌ها را محاسبه نموده و نمایش دهد.