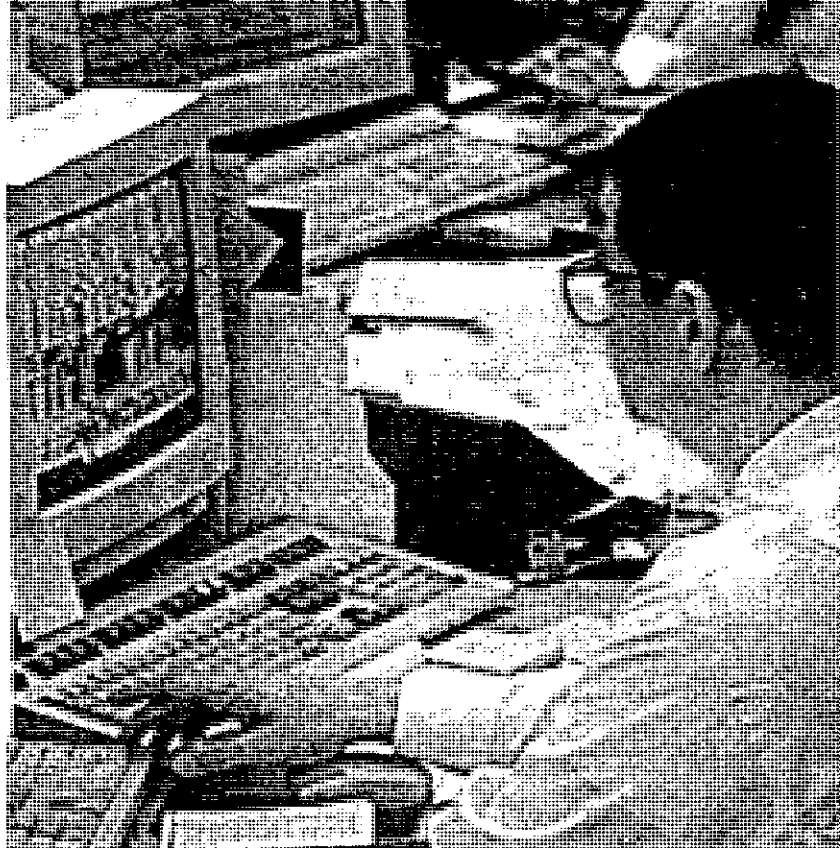


تفکر الگوریتمی، هنر برنامه‌نویسی

● نوشتهٔ عبدالحسین مصحفی



حل ریاضی یک مسأله با الگوریتم حل کامپیوتری آن تفاوت‌هایی دارد. ابزارهایی که در این دو الگوریتم به کار می‌روند با یکدیگر متفاوتند. این بدان معنی نیست که در راه حل کامپیوتری باید راه حل ریاضی را نادیده گرفت، بلکه نخست باید راه حل ریاضی را به دست آورد و آنگاه با به کار بردن ابزارها و ساختارهای نرم‌افزاری، آن راه حل ریاضی را به صورت الگوریتمی کامپیوتری درآورد. الگوریتم حل مسأله که معلوم شود نوشتن آن به یکی از زبانهای برنامه‌نویسی کاری به نسبت ساده خواهد بود، به پاکتویس کردن یک انشای طرح‌ریزی شده می‌ماند.

هنر برنامه‌نویسی در آن است که متناسب با مسأله موردنظر، کاراترین ابزارها و ساختارهای نرم‌افزاری گزیده شوند و کوتاهترین و زیباترین الگوریتم فراهم آید. از این رو برای ورزیده شدن در هنر برنامه‌نویسی، شناسایی ابزارها و ساختارهای نرم‌افزاری و بی‌پردن به چگونگی کارکرد آنها نخستین گام است. گامهای بعدی، تمرین هرچه بیشتر و به دست آوردن الگوریتمهای مربوط به حل مسأله‌هایی از گونه‌های متفاوت است.

ابزارها و ساختارهای نرم‌افزاری

متغیرها: در برنامه‌نویسی، متغیرها، همانند متغیرهای جبری، برای کلیت دادن به دستورها و قاعده‌ها به کار می‌روند. اما در

با به پای گسترده‌تر شدن کاربردهای کامپیوتر، گرایش به ورزیدگی در کار کردن با آن نیز افزوتر شده است. در این باره سروکار هم با سخت‌افزار است و هم با نرم‌افزار. شناخت سخت‌افزار، بیشتر جنبهٔ حرفه‌ای دارد و به سادگی برگزار می‌شود. در زمینهٔ نرم‌افزار نیز برنامه‌نویسی مسأله‌هایی که جنبهٔ استاندارد داشته باشند، دشواری ویژه‌ای را در پی نخواهد داشت؛ کافی است که قالب برنامهٔ مربوط به هر مسأله شناخته شود و به کار رود. برای برنامه‌نویسی، مسأله‌هایی از گونه‌های دیگر، از جمله بسیاری از مسأله‌های ریاضی، تفکری ویژه لازم است که تفکر الگوریتمی یا هنر برنامه‌نویسی نامیده می‌شود.

دانش‌آموزانی که در مسابقه‌ها و در المپیادهای کامپیوتری می‌درخشند، از چنین تفکری برخوردارند و در چنین هنری ورزیدگی دارند. دانش‌آموزان و کسانی دیگر هم دستیابی به روشهایی را خواهند که پیشرفت و ورزیده شدن آنها را در برنامه‌نویسی در پی داشته باشد. امید است که این نوشتار برای اینان کارساز باشد.

برای نوشتن برنامهٔ مربوط به راه حل یک مسأله، نخستین گام، بی‌پردن به الگوریتمی است که باید به کار برد. الگوریتم، که تحریف شدهٔ الخوارزمی است، و امروزه به معنی ویژهٔ «راه حل کامپیوتری» زیانزد شده، همان اصطلاح «قاعده (روش)» است که در متنهای ریاضی به کار می‌رفته و به کار می‌رود. اما الگوریتم

زیروند داشته باشد، آرایه را یک بعدی، دو بعدی، ... یا n بعدی می‌نامند. آرایه یک بعدی یک دنباله و آرایه دو بعدی یک جدول یا یک ماتریس خواهد بود. دنباله رقمهای عدد نویسی دهدهی آرایه‌ای یک بعدی می‌سازند که اگر نام آن را D بگذاریم هر درایه‌اش (= هر رقم) با $D(I)$ نموده می‌شود که I یکی از شماره‌های از ۰ تا ۹ است. جدول ضرب دهدهی آرایه‌ای دو بعدی است که با انتخاب مثلاً MUL برای نام آن، هر درایه‌اش با $MUL(I,J)$ نموده می‌شود که I شماره سطر و J شماره ستون است، چنان‌که $MUL(7,3)$ نمایانگر 7×3 و برابر با ۲۱ است.

در هریک از زبانهای برنامه‌نویسی، دستوری ویژه برای معرفی آرایه وجود دارد. آرایه‌ای یک بعدی از عددهای صحیح که A نام و ۵۰ اندازه آن باشد، در زبان BASIC به صورت

`DIM A(50)`

در زبان PASCAL به صورت

`TYPE A = ARRAY[0 .. 50] OF INTEGER;`

`int a [50]`

و در زبان C به صورت

معرفی می‌شود.

همانند متغیرهای ساده، برای هر آرایه نیز کامپیوتر خانه‌هایی به تعداد درایه‌های آن را در حافظه خود ذخیره می‌کند. هر کدام از این خانه‌ها با متغیر زیرونددار، نظیر هریک از درایه‌های آرایه نشان می‌شوند و مقدار آغازی هر کدام صفر خواهد بود.

گمارش و جایگزین کردن: یک مقدار یا حاصل عددی یک عبارت جبری را می‌توان به یک متغیر نسبت داد (در آن گمارد) یا اینکه آن را به جای مقدار قبلی آن متغیر قرار داد (آن را جایگزین مقدار قبلی متغیر کرد) در این مورد فرق نمی‌کند که متغیر ساده یا زیرونددار باشد. در بیشتر زبانهای برنامه‌نویسی همان نشانه « $=$ » و در زبان پاسکال نشانه « $:=$ » را برای هریک از دو عمل گمارش و جایگزینی به کار می‌برند؛ نام متغیر در سمت چپ نشانه و مقدار ثابت یا عبارتی که حاصل آن باید در آن متغیر جایگزین شود در سمت راست نشانه نوشته می‌شود. دستور گمارش و جایگزینی در زبان بیسیک بیشتر با `LET` آغاز می‌شد و امروزه می‌توان از به کار بردن آن خودداری کرد. هریک از دو دستور

`LET PI=3.14` یا `PI=3.14`

از زبان بیسیک و دستور

`PI:=3.14`

کامپیوتر هرگونه محاسبه و پردازش روی مقدارهای معلوم انجام می‌گیرد. اگر در جبر می‌توان عملهایی را روی خود متغیرها انجام داد، مثلاً حاصل عبارت $(2x+1)^2$ را برحسب x به دست آورد، در کامپیوتر حاصل این عبارت، تنها آنگاه به دست می‌آید که معلوم شده باشد چه مقدار عددی باید جانشین x شود و حاصل عمل هم یک مقدار عددی خواهد بود.

هر متغیر با یک نام نموده می‌شود. این نام می‌تواند یک حرف، ترکیبی بی‌معنی یا با معنی از یک یا چند حرف و عدد باشد.

چنان‌که نامهایی از گونه `SUM`، `DICE`، `X1`، `COUNT` را می‌توان برای متغیرها برگزید. به هنگام اجرای برنامه، کامپیوتر با هر متغیر که روبرو شود جایی (خانه‌ای) از حافظه خود را برای آن متغیر در نظر می‌گیرد و هرگونه پردازش روی آن متغیر را روی مقدار موجود در آن جای انجام می‌دهد. مقدار یک متغیر اگر معلوم نشده باشد، کامپیوتر آن را صفر می‌گیرد.

متغیرهای زیرونددار: همان‌گونه که در ریاضیات متغیر زیرونددار، مثلاً x_n برای نشان دادن مجموعه مقدارهایی به کار می‌رود که همه این مقادارها با نام x شناسایی و با زیروندهای ۱، ۲، ۳، ...، n ، از یکدیگر تمیز داده می‌شوند، در الگوریتمها و برنامه‌نویسیها نیز می‌توان مجموعه مقدارهایی را با یک متغیر زیرونددار نشان داد و به کار برد. اما در اینجا زیروند، که عدد صحیح نامنفی است، در کنار متغیر و داخل پرانتز یا گروه نوشته می‌شود. چنان‌که متغیر $X(1)$ یا $X[I]$ به معنی متغیر با نام X و با زیروند ۱ است در حالی که $X1$ نشانه یک متغیر ساده است. متغیرهای با دو زیروند، با سه زیروند و با زیروندهای بیشتر را می‌توان به کار برد.

آرایه: مقدارهایی (عددی یا غیر عددی) که چنان مرتب شده باشند که بتوان آنها را شماره‌گذاری کرد و به هریک از آنها از روی شماره‌اش پی‌برد، مجموعه‌ای منظم را تشکیل می‌دهند که آن را آرایه، هریک از آن مقادارها را یک درایه (عضو) آن آرایه، و تعداد درایه‌ها را اندازه آن آرایه می‌نامند. هر آرایه با یک نام و با اندازه‌اش شناخته می‌شود و هر درایه آن را می‌توان با یک متغیر زیرونددار نشان داد. این متغیر همان نام آرایه و زیروند آن شماره ترتیب درایه است. در بیشتر زبانهای برنامه‌نویسی، شماره‌گذاری درایه‌های آرایه از صفر آغاز می‌شود. بنابر آن که متغیر نمایانگر درایه‌های آرایه، یک زیروند، دو زیروند، ... یا n

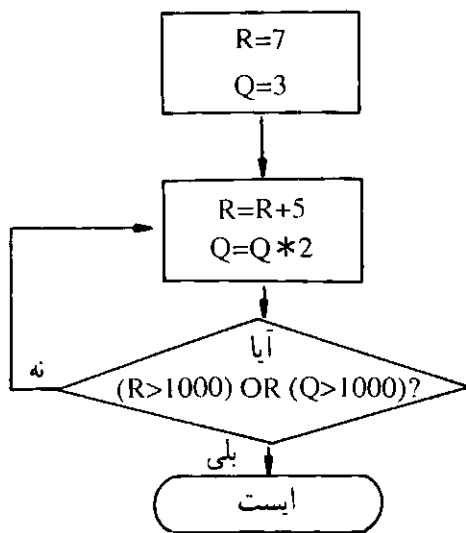
بنابراین فرایند، دو دنباله‌ی پایان زیر پدید می‌آیند:

۷, ۱۲, ۱۷, ۲۲, ...

۳, ۶, ۱۲, ۲۴, ...

که اولی تصاعد حسابی و دومی تصاعد هندسی است.

گزینش: بنابر نمودار شکل (۱) تکرار دو عمل جایگزینی بدون وقفه ادامه خواهد داشت. اما در عمل چنین امکانی وجود ندارد و لازم است که در مرحله‌ای از تکرار عملها جلوگیری شود. در این مرحله، شرطی گنجانیده می‌شود که بنابر آن که برقرار باشد یا نباشد. عمل تکرار یا متوقف می‌شود یا باز دنبال می‌شود. چنین شرطی یک مرحله گزینش را بر سر راه قرار می‌دهد که باید از دو یا چند امکان تنها یکی از آنها برگزیده شود. برای مثال، اگر توقف عمل تکرار در نمودار شکل (۱) موکول به آن باشد که دست کم یکی از دو مقدار R یا Q از ۱۰۰۰ تجاوز کند، در این صورت نمودار شکل (۲) را خواهیم داشت:



(شکل ۲)

بنابراین نمودار، پس از مرحله جایگزینی یک مرحله گزینش (= یک دوراهی) وجود دارد؛ هرگاه شرط بیشتر شدن یکی از دو مقدار R یا Q از ۱۰۰۰ برقرار شده باشد، عمل پایان می‌یابد، وگرنه بازهم عملهای جایگزینی تکرار می‌شوند.

در بیشتر الگوریتمها، دست کم یک مرحله گزینش وجود دارد که در آن مسیر گردش کار دو یا چند شاخه می‌شود و بنابر آنچه مورد نظر باشد، یکی از مسیرها گزیده می‌شود.

چرخه (=حلقه): در بیشتر الگوریتمها لازم می‌شود که یک عمل یا مجموعه‌ای از چند عمل یا به تعداد معین یا تا برقراری

از زبان پاسکال، به معنی گماشتن عدد ۳/۱۴ در متغیر به نام PI است.

در دستور جایگزینی ممکن است که عبارت طرف راست نشانه همان متغیر سمت چپ را نیز دربرداشته باشد. در این صورت، حاصل عبارت طرف راست، برحسب آخرین مقدار آن متغیر حساب می‌شود و آنچه به دست آید، مقدار جدید آن متغیر می‌شود. چنان که اگر $F(A, X, SUM)$ عبارتی جبری برحسب متغیرهای A, X و SUM باشد، دستور

$$A = F(A, X, SUM)$$

به این معنی است که:

(مقدار SUM و مقدار X و مقدار قبلی A) = مقدار جدید A

مثال ۱: در برنامه‌ای به زبان بیسیک، هرگاه سه مرحله گمارشها و جایگزینی‌های زیر یکی پس از دیگری انجام گیرند:

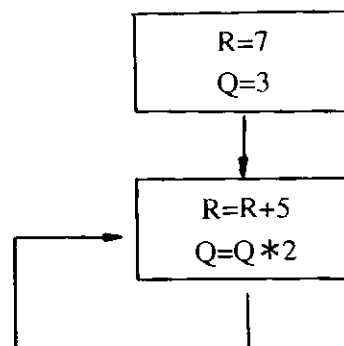
$$(۱): I=10, J=17, A(I, J) = -20$$

$$(۲): I=I+1, J=J+1, A(I, J) = A(I-1, J-1) + 7$$

$$(۳): I=5, J=J-4, A(I, J) = 2 * (I+J)$$

در مرحله (۱)، مقدار I برابر ۱۰، مقدار J برابر ۱۷ و مقدار A(۱۰ و ۱۷) از آرایه A برابر ۲۰- می‌شود. در مرحله (۲)، مقدار I برابر با ۱۱ و ۱۰+۱=۱۱، مقدار J برابر با ۱۸ و ۱۷+۱=۱۸ و مقدار A(۱۱ و ۱۸) برابر با ۱۳- می‌شود. در مرحله (۳)، مقدار I برابر با ۵، مقدار J برابر با ۱۴ و ۱۸-۴=۱۴ و مقدار A(۵ و ۱۴) برابر با ۳۸ می‌شود. (نشانه * به جای عملگر ضرب به کار می‌رود).

مثال ۲: بنابر نمودار شکل (۱)، مقدار R در آغاز برابر ۷ و مقدار Q در آغاز برابر ۳ است. پس از آن رابطه‌های جایگزینی $R=R+5$ و $Q=Q*2$ پشت سرهم تکرار می‌شوند که در هر بار تکرار آنها مقدار R افزایشی به اندازه ۵ خواهد داشت و مقدار Q دوبرابر می‌شود.



(شکل ۱)

تشکیل می‌شود که هر کدام از این بخشها ممکن است، بخشهای کوچکتری را دربرداشته باشد. در نخستین بخش از الگوریتم، چگونگی دسترسی به داده‌هایی که باید به کار روند مشخص می‌شود و متغیرهای ساده یا زیرونداری که باید داده‌ها را در خود نگهدارند انتخاب و معرفی می‌شوند. این بخش از الگوریتم که به زبان بیسیک برگردانده شود، ساختاری بسیار ساده دارد. دومین بخش اصلی الگوریتم، که معمولاً چند بخش فرعی را دربر دارد، از ابزارها و ساختارهایی نرم‌افزاری تشکیل می‌شود که برای پردازشهای لازم روی داده‌ها به کار می‌روند. در این پردازشها گاه لازم می‌شود متغیرهایی کمکی نیز به کار روند؛ وقتی روی متغیری عمل جایگزینی انجام گیرد مقدار قبلی آن متغیر از دسترس خارج می‌شود، حال اگر بنا باشد روی مقدار قبلی بازهم پردازشی انجام گیرد از متغیر کمکی برای نگهداری آن مقدار استفاده می‌شود. در سومین بخش اصلی الگوریتم، چگونگی در دسترس گذاشتن پاسخ یا پاسخهای خواسته شده معلوم می‌شود.

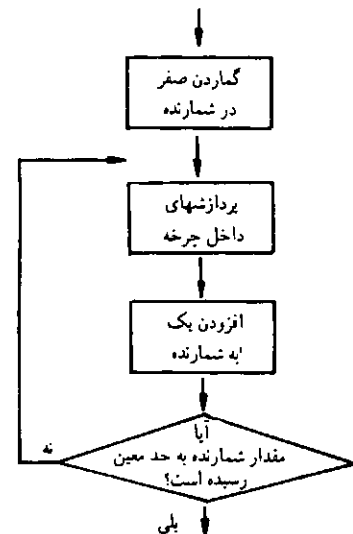
الگوریتم پیش از نموده شدن، در ذهن، طرح‌ریزی و ساختار هر یک از بخشها و ترتیب به کار رفتن آنها مشخص می‌شود. پس از نموده شدن الگوریتم هم باید مقدارهایی معلوم روی آن آزمایش شوند و درستی الگوریتم بررسی گردد. یک الگوریتم خوب، آن است که برنامه نظیر آن برای بار شدن روی کامپیوتر جایی هرچه کمتر لازم داشته باشد و برای اجرای آن زمانی هرچه کوتاهتر صرف شود. برای این کار، باید تعداد متغیرهای به کار رفته در آن و هم تعداد عملهای پردازشی آن، هرچه ممکن است کمتر باشند. نه تنها هر چرخه به کار رفته در الگوریتم، بلکه خود الگوریتم نیز باید در مرحله‌ای پایان یابد.

مسأله نمونه

می‌خواهیم الگوریتم برنامه‌ای را طرح کنیم که بنابر آن برنامه، هر بار که یک عدد حقیقی (در حالت کلی به صورت ممیزدار و با تعداد محدودی از رقمها) به کامپیوتر داده شود معلوم کند هر یک از رقمهای دهگانه چند بار به صورت با معنی در آن عدد به کار رفته است (صفرهای واقع در سمت چپ عدد صحیح و صفرهای واقع در سمت راست کسر دهمی رقمهای با معنی به شمار نمی‌آیند).

متغیر N را انتخاب می‌کنیم تا عددی را که هربار به کامپیوتر داده می‌شود، در خود نگه‌دارد. الگوریتم باید هربار برای هر یک

شرطی معین تکرار شود. چنین فرآیندی را چرخه عملیاتی (=حلقه عملیاتی) و بیشتر فقط چرخه یا حلقه می‌نامند. چرخه‌ها از مهمترین ساختارهای نرم‌افزاری‌اند و یک ابزار توانمند الگوریتمها به شمار می‌آیند. هر چرخه یک مرحله گزینش دارد که در آن یک شرط بررسی می‌شود و بنابر آن، یا گردش کار چرخه پایان می‌یابد یا باز تکرار می‌شود. این شرط ممکن است یک گزاره (ساده یا مرکب) باشد. مانند شرط شکل (۲) که ترکیب فصلی دو گزاره است، و ممکن است که شرط توقف چرخه یک شمارنده باشد. در این حالت، یک متغیر به نام شمارنده انتخاب و پیش از آغاز کار چرخه مقدار آن صفر (یا یک، بنابر آن که مرحله گزینش در آغاز یا در پایان چرخه باشد) می‌شود و با هر بار تکرار چرخه، یک بار عدد یک به شمارنده افزوده می‌شود. در مرحله گزینش بررسی می‌شود که آیا مقدار شمارنده به حد مورد نظر رسیده است یا نه. ساختار کلی یک چرخه شمارشی مطابق با نمودار شکل (۳) است:



(شکل ۳)

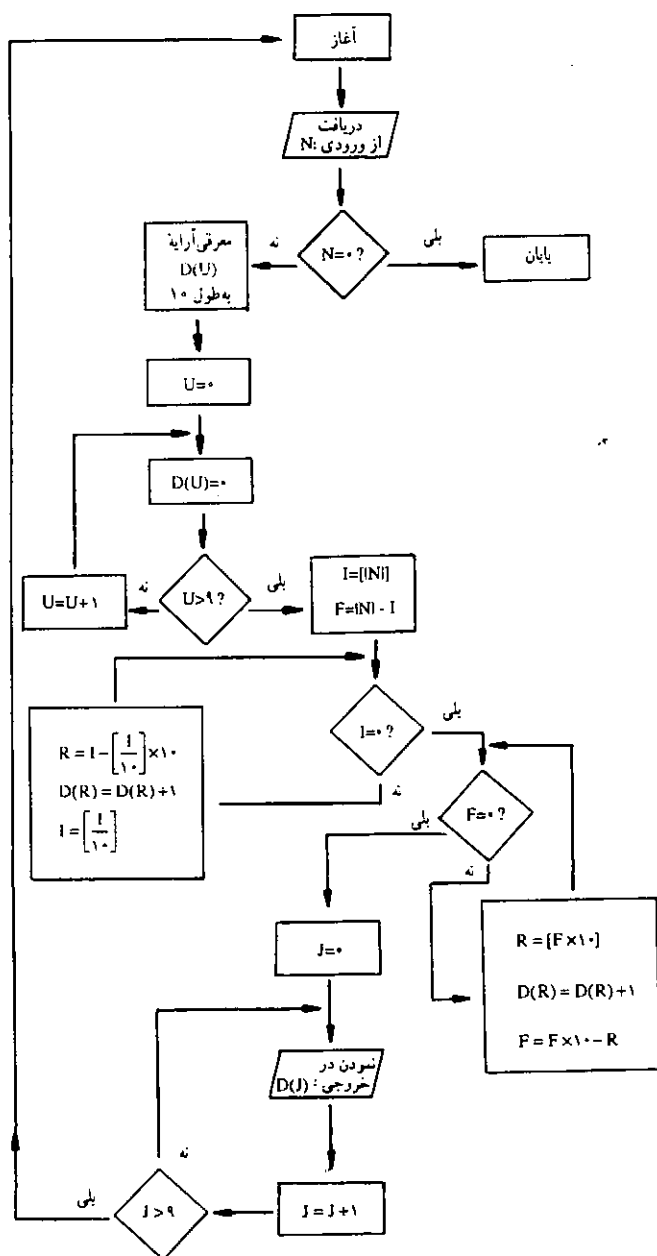
در هر یک از زبانهای برنامه‌نویسی، دستور عملهایی ویژه هریک از دو گونه چرخه وجود دارد. چنان که در زبان بیسیک دستور عملهای FOR - NEXT را برای چرخه شمارشی و دستور عملهای WHILE - WEND را برای چرخه شرطی می‌توان به کار برد. بدون به کار بردن این دستور عملهای ویژه، می‌توان الگوریتم شکل (۳) را به زبان بیسیک برگرداند.

ساختار الگوریتم

هر الگوریتم یک برنامه کامپیوتری معمولاً از سه بخش اصلی

پردازشهای مربوط به، به دست آوردن و شمارش رقمها که پایان یابند، مقدار واقع در هر درایه از آرایه D برابر با تعداد رقمهای برابر با زیروند آن درایه است. این مقادارها با یک چرخه شمارشی در خروجی دستگاه نموده می شوند.

الگوریتم بیان شده، مطابق با نمودار گردش کار شکل (۴) و برنامه آن به زبان بیسیک می تواند به گونه ای باشد که در صفحه پس از آن درج می شود.



(شکل ۴)

از این عددها تکرار شود و برای باز ایستادن این چرخه تکرار، عدد صفر را به کار می بریم؛ هرگاه عدد صفر به کامپیوتر داده شود، روی آن پردازشی انجام نگیرد و الگوریتم پایان یابد.

برای شمارش هریک از رقمهای به کار رفته، باید ده متغیر را به عنوان شمارنده به کار ببریم. این ده شمارنده را با متغیر زیرونددار D(U) نشان می دهیم که U یکی از عددهای از ۰ تا ۹ خواهد بود. این متغیر زیرونددار به صورت آرایه با نام D و به اندازه ۱۰ معرفی می شود و مقدار آغازی هریک از درایه های آن صفر می گردد. شمارش رقمها به این ترتیب انجام می گیرد که با هر بار به دست آمدن یکی از آنها، یک بار عدد یک به مقدار درایه با زیروند برابر با آن رقم افزوده می شود. مثلاً هر بار که رقم ۷ به دست آید، یک به مقدار D(۷) افزوده می شود.

برای به دست آوردن رقمها، در مرحله نخست، قدر مطلق جزء صحیح عدد را در متغیری به نام I و قدر مطلق جزء کسری را در متغیری به نام F می گماریم که ممکن است مقدار یکی از اینها صفر باشد. جزء صحیح را از راه به کار بردن تابع جزء صحیح، که در زبان بیسیک با INT(X) نموده می شود، و جزء کسری را از کم کردن جزء صحیح از خود عدد به دست می آوریم. تابع جزء صحیح همان است که در ریاضیات با [X] نشان داده می شود و کوچکترین عدد صحیح نابزرگتر از عدد مفروض است. از آن رو که اگر عدد منفی باشد علامت منفی در نوع رقمهای عدد تأثیری ندارد، قدر مطلق عدد مفروض را در نظر می گیریم.

پس از جدا کردن جزءهای صحیح و کسری عدد از هم، جزء صحیح را که بر ۱۰ تقسیم کنیم، خارج قسمت به دست آمده را باز بر ۱۰ تقسیم کنیم، و این فرآیند را تا زمانی تکرار کنیم که خارج قسمت صفر به دست آید، باقیمانده های تقسیمها رقمهای جزء صحیح عدد (به ترتیب از راست به چپ) خواهند بود. همچنین اگر جزء کسری عدد را در ۱۰ ضرب کنیم و جزء کسری حاصل ضرب را باز در ۱۰ ضرب کنیم و این فرآیند را تا زمانی تکرار کنیم که حاصل ضرب با جزء کسری برابر با صفر به دست آید، جزءهای صحیح حاصل ضربها رقمهای جزء کسری عدد مفروض (به ترتیب از چپ به راست) خواهند بود. هر بار که یک تقسیم بر ۱۰ انجام گیرد، خارج قسمت به جای مقسوم در I جایگزین می شود، همچنین پس از هر بار عمل ضرب، جزء کسری حاصل ضرب جایگزین مقدار قبلی F می شود. به این ترتیب هر دو گونه پردازش همواره روی دو متغیر I و F انجام می گیرد.

مسئله‌هایی برای تمرین

الگوریتم حل کامپیوتری هر یک از مسئله‌های زیر را طرح کنید و آن را یا با نمودار نشان دهید یا به یکی از زبانهای برنامه‌نویسی بنویسید :

۱ - دنباله عددی $[I_n]$ به گونه زیر تعریف شده است :

$$I_1 = 1 \text{ و } I_n = n + 2I_{n-1}$$

جمله‌های این دنباله و شماره ترتیب آنها تا آنگاه به دست آیند که آخرین جمله از ۱۰۰۰ بزرگتر نباشد.

۲ - کسری متعارفی ساده نشدنی و کوچکتر از یک به دهمی دوره‌ای ساده تبدیل شده است. عدد A برابر با بخشی (ابتدا از ممیز) این کسر دوره‌ای است که دست کم یک دوره گردش را دربر دارد و عدد طبیعی N برابر با تعداد رقمهای دوره گردش است. دو عدد A و N داده می‌شوند. صورت و مخرج کسر متعارفی مفروض را به دست آورید.

۳ - در مسئله دانه‌های گندم «صفحه شطرنج» جواب برابر با «۱-۲۶۴» است. به فرض آنکه هر عدد صحیح بزرگتر از $۶۵۵۳۶ = ۲^{۱۶}$ در کامپیوتر به صورت نمایی نموده شود که بنابر آن، تعدادی از رقمهای غیر صفر عدد به صفر تبدیل می‌شوند. چه الگوریتمی را باید به کار برد تا بنابر آن، تعداد دانه‌های گندم مسئله صفحه شطرنج با همه رقمهای اصلی خود در خروجی کامپیوتر نموده شود؟



برنامه به زبان بیسیک الگوریتم نمودار شکل (۴) می‌تواند چنین باشد :

```

10 INPUT N
20 IF N=0 THEN 200
30 DIM D (10)
40 FOR U=0 TO 9
50 D(U)=0 : NEXT U
60 I=INT (ABS(N)) : F=ABS(N)-I
70 IF I=0 THEN 110
80 R=I-INT (I/10)* 10
90 D(R)=D(R)+1
100 I=INT(I/10) : GOTO 70
110 IF F=0 THEN 150
120 R=INT (F*10)
130 D(R)=D(R)+1
140 F=F*10-R : GOTO 110
150 PRINT "Each of the digits has been used in
number as follow:"
160 FOR J=0 TO 9
170 PRINT "The digit" ; J,D(J) "times"
180 NEXT J
190 GOTO 10
200 END

```

برای مثال می‌توانید عدد $۸۹۰۸۷/۳۵۹$ را به جای N در نظر بگیرید و آن را روی برنامه آزمایش کنید که خروجی چنین باشد : ۱ بار رقم ۰، ۰ بار رقم ۱، ۰ بار رقم ۲، ۱ بار رقم ۳، ۰ بار رقم ۴، ۱ بار رقم ۵، ۰ بار رقم ۶، ۱ بار رقم ۷، ۲ بار رقم ۸، ۲ بار رقم ۹.