الگوریتم-فلوچارت

و

مبانی برنامه‌نویسی

به زبان C++

(کتاب درسی برای دانشجویان فنی و مهندسی و علوم پایه)

تالیف: **دکتر جواد شکری**

عضو هیأت علمی گروه ریاضی، دانشکده علوم پایه

دانشگاه ارومیه

سرشناسه:	**شکری، جواد، ۱۳۵۱ -**
عنوان و نام پدیدآور:	**الگوریتم - فلوچارت و مبانی برنامه‌نویسی به زبان ++C / تالیف جواد شکری.**
مشخصات نشر:	**ارومیه: دانشگاه ارومیه، انتشارات، ۱۴۰۲.**
مشخصات ظاهری:	**۴۵۶ ص.**
فروست:	**انتشارات دانشگاه ارومیه؛ ۳۲۲.**
شابک:	978-622-5791-16-9
وضعیت فهرست نویسی:	**فیپا**
یادداشت:	**واژه‌نامه.**
یادداشت:	**کتابنامه ص. ۴۵۵**
عنوان دیگر:	**سی ++ (زبان برنامه‌نویسی کامپیوتر) C++) (Computer program language)**
موضوع:	**الگوریتم‌های کامپیوتری -- راهنمای آموزشی Computer algorithms -- Study and teaching**
موضوع:	**برنامه‌نویسی -- راهنمای آموزشی Computer programming -- Study and teaching**
موضوع:	**نمودار جریان کار -- راهنمای آموزشی Flow charts -- Study and teaching**
شناسه افزوده:	**دانشگاه ارومیه. انتشارات**
رده بندی کنگره:	**QA ۷۶/۷۳**
رده بندی دیویی:	**۱۳۳/۰۰۵**
شماره کتابشناسی ملی:	**۹۰۹۵۶۵۶**

مرکز انتشارات دانشگاه ارومیه

ارومیه، کیلومتر ۱۱ جاده سرو، صندوق تلفن: ۳۱۹۴۲۲۷۴ -۳۲۷۷۹۹۳۰ -۰۴۴ ،دورنگار ۳۲۷۷۹۹۳۰

عنوان: الگوریتم - فلوچارت و مبانی برنامه‌نویسی به زبان ++C

تالیف: دکتر جواد شکری

ویراستار علمی: دکتر اسماعیل نجفی

ناشر: انتشارات دانشگاه ارومیه

لیتوگرافی، چاپ و صحافی: طاها نگار

نوبت چاپ: اول

سال چاپ:۱۴۰۲

شمارگان: ۵۰۰ نسخه

قیمت پشت جلد:۳۱۰۰۰۰ تومان

شابک: ۹۷۸-۶۲۲-۵۷۹۱-۱۶-۹

بسم الله الرحمن الرحيم

تقدیم به:

ارواح بلند

پدر بزرگوار و مادر مهربانم

پیش‌گفتار

نقش چشمگیر کامپیوتر بر زندگی روزمره بشر به‌وضوح قابل مشاهده است. اما کامپیوتر با وجود کاربردهای گوناگون، بدون برنامه‌نویسی یا به بیان دیگر بدون دستورالعمل‌های دقیق و قدم به قدم قادر به انجام محاسبات یا پردازش داده‌ها نمی‌باشد. مجموعه این دستورالعمل‌ها که بیانگر راه حل قدم به قدم مسئله‌ای یا نشانگر ترتیب توالی انجام مراحل یک عمل یا رویداد می‌باشند، برنامه کامپیوتری نامیده می‌شود.

نوشتن برنامه در مرحله سوم از فرایند برنامه‌سازی انجام می‌گیرد و قبل از آن در مرحله اول تعریف مسئله و معلوم نمودن روابط محاسباتی لازمه مسئله می‌باشد، سپس در مرحله دوم باید منطق برنامه طراحی گردد. منطق برنامه به صورت کلامی یا به صورت نمایش تصویری می‌باشد.

یکی از روش‌های بیان منطق برنامه به صورت کلامی ارائه آن به صورت الگوریتم است. الگوریتم از نام خوارزمی(الخوارزمی در زبان عربی) دانشمند بزرگ ایرانی در قرن دوم هجری قمری گرفته شده است. خوارزمی برای بار اول توانست حل یک مسئله ریاضی را به صورت گام به گام بیان نماید. نمایش تصویری بیان منطق برنامه، فلوچارت یا رَوَندنما می‌باشد که در این روش دقت و پیوستگی یک الگوریتم را به کمک اشکال هندسی به صورت یک طرح یا دیاگرام توأم می‌سازد.

منطق برنامه که شامل الگوریتم و فلوچارت برنامه می‌باشد، شالوده و اساس برنامه‌سازی کامپیوتر است که در شرکت‌های برنامه‌نویسی کامپیوتر به این گروه از افراد اتاق فکر برنامه‌سازی گفته می‌شود. به علت اهمیت مسئله است که *الگوریتم و نظریه محاسبه* یکی از شاخه‌های تحصیلی در دوره تحصیلات تکمیلی رشته علوم کامپیوتر است.

لذا بنا به اهمیت و ضرورت آموختن طراحی منطق برنامه در امر برنامه‌نویسی کامپیوتر، ما در این کتاب سعی نموده‌ایم با رعایت این اصول، کتاب را در دو بخش طراحی نماییم. ابتدا در بخش اول منطق برنامه، شامل الگوریتم و فلوچارت برنامه‌های مختلف، با رعایت موضوع، ارائه گردیده است، سپس در بخش دوم، برنامه‌نویسی به زبان C++ ارائه خواهد شد.

بخش منطق برنامه شامل شش فصل می‌باشد که در فصل ۱ مقدمه‌ای بر مفاهیم الگوریتم و فلوچارت برنامه ارائه شده است و در همین فصل با ارائه منطق چند برنامه سعی شده است این مفاهیم تشریح گردد. در فصل‌های ۲ الی ۶ بخش الگوریتم و فلوچارت به‌ترتیب مربوط به برنامه‌های حاوی ساختارهای متوالی، ساختارهای کنترلی انتخاب، ساختارهای تکرار، زیرالگوریتم‌ها و آرایه می‌باشد. در تمامی مباحث این بخش سعی شده است که برنامه‌ها از ساده به دشوار طراحی گردند.

بخش برنامه‌نویسی به زبان C++ شامل پنج فصل، از فصل ۷ الی فصل ۱۱، می‌باشد. فصل ۷ شامل مفاهیم اولیه زبان برنامه‌نویسی C++ است، و در ادامه فصل ۸ شامل ساختارهای کنترلی، فصل ۹ شامل مبحث توابع، فصل ۱۰ شامل آرایه و در نهایت فصل ۱۱ شامل مبحث اشاره‌گرها می‌باشد. در پایان هر فصل چندین برنامه مفید به همراه خروجی آنها ارائه شده است.

یکی از قابلیت‌های مشخصه زبان C++ شیءگرایی این زبان است که اشاره‌گرها سرآغاز مفاهیم مربوط به این قابلیت است که در فصل ۱۱ کتاب آورده شده است. بنا به هدف و ماهیت این کتاب مباحث شیءگرایی در قالب این کتاب نمی‌گنجد، لذا علاقه‌مندان این مبحث را در برنامه‌نویسی پیشرفته دنبال خواهند نمود.

بنا به تجربه چندساله‌ای که نویسنده در تدریس این درس با عنوان مبانی کامپیوتر و برنامه‌سازی داشته است به عرض اساتید گرامی این ماده درسی، با فرض ۳۰ جلسه در ترم تحصیلی، می‌رساند که با اطمینان خاطر می‌توانند حدود ۱۲ جلسه به آموزش بخش الگوریتم و فلوچارت برنامه‌ها تخصیص دهند، سپس با شروع بخش برنامه‌نویسی ملاحظه خواهد شد که دانشجو به‌سهولت و در زمان کمتر از مورد انتظار برنامه‌نویسی را با مهارت کافی خواهد آموخت و البته بهتر می‌دانیم که برنامه‌نویسی بدون کار عملی نتیجه‌بخش نخواهد بود.

با آرزوی پیروزی روزافزون شما در راه علم و دانش امیدوارم با پیشنهادهای سازنده خود نویسنده کتاب را در هرچه بهتر نمودن این اثر یاری فرمایید.

جواد شکری

شهریور ۱۴۰۲

فهرست مطالب

عنوان صفحه

فصل ۱. مفاهیم الگوریتم و فلوچارت ۱

۱-۱. مفهوم الگوریتم ۱

۱-۲. ویژگی‌های یک الگوریتم..... ۳

۱-۳. زیرالگوریتم ۵

۱-۴. فلوچارت یا رَوَندنما ۶

۱-۵. نمادهای مورد استفاده در فلوچار..... ۷

۱-۶. رسم فلوچارت‌های نمونه ۸

تمرین ۱۱

فصل ۲. الگوریتم و فلوچارت ساختارهای متوالی ۱۳

تمرین ۲۷

فصل ۳. الگوریتم و فلوچارت ساختارهای کنترلی انتخاب ۲۹

تمرین ۵۱

فصل ۴. الگوریتم و فلوچارت حاوی ساختارهای تکراری ۵۳

تمرین ۱۰۷

فصل ۵. زیرالگوریتم‌ها و فلوچارت‌های متناظر آنها ۱۰۹

۵-۱. زیرالگوریتم‌ها ۱۰۹

۵-۲. زیرالگوریتم‌های بازگشتی..... ۱۲۰

تمرین ۱۲۵

فصل ۶. الگوریتم‌ها و فلوچارت‌های حاوی آرایه ۱۲۷

۶-۱. الگوریتم‌ها و فلوچارت‌های حاوی آرایه یک بعدی ۱۲۷

۶-۲. الگوریتم و فلوچارت مرتب‌سازی آرایه‌ها ۱۳۴

۶-۳. الگوریتم و فلوچارت جست‌وجو در آرایه‌ها ۱۴۷

۶-۴. الگوریتم‌ها و فلوچارت‌های حاوی آرایه دو بعدی ۱۵۵

تمرین ۱۶۷

فصل ۷. مفاهیم اولیه زبان برنامه‌نویسی C++ ۱۶۹

۷-۱. بعضی از ویژگی‌های زبان C++ ۱۷۰

۷-۲. انواع داده‌ها ۱۷۱

۷-۳. شناسه ۱۷۲

۷-۴. واژه‌های کلیدی ۱۷۳

۷-۵. اعلان متغیرها ۱۷۴

۷-۶. اعلان ثابت‌ها ۱۷۶

۷-۷. عملگرها ۱۷۷

۷-۸. تبدیل انواع داده‌ها ۱۸۷

۷-۹. روش ایجاد برنامه ساخت‌یافته ۱۸۸

۷-۱۰. ساختار برنامه در C++ ۱۹۰
۷-۱۱. دستورات ورودی و خروجی ۱۹۳
۷-۱۲. چند تابع ریاضی ۲۰۱
تمرین ۲۰۵
ارائه چند برنامه مفید ۲۰۶
فصل ۸. ساختارهای کنترلی ۲۱۷
۸-۱. ساختار تصمیم ۲۱۷
۸-۲. ساختار تکرار for ۲۲۳
۸-۳. ساختار تکرار while ۲۳۵
۸-۴. ساختار تکرار do-while ۲۳۹
۸-۵. ساختار تکرار switch ۲۴۵
تمرین ۲۵۱
ارائه چند برنامه مفید ۲۵۵
فصل ۹. توابع و کلاس‌های حافظه ۲۷۳
۹-۱. توابع و برنامه‌سازی ساخت‌یافته ۲۷۳
۹-۲. نوشتن توابع ۲۷۴
۹-۳. روش‌های ارسال پارامترها به توابع ۲۷۷
۹-۴. توابعی که هیچ مقداری را برنمی‌گردانند ۲۷۸
۹-۵. توابعی که یک مقدار را برمی‌گردانند ۲۸۲
۹-۶. توابع بازگشتی ۲۸۷

۹-۷. متغیرهای محلی و عمومی ۲۹۰
۹-۸. توابع inline ۲۹۲
۹-۹. قالب‌های توابع ۲۹۴
۹-۱۰. نوع داده enum ۲۹۶
۹-۱۱. کلاس‌های حافظه یک متغیره ۲۹۹
۹-۱۲. typedef ۳۰۳
تمرین ۳۰۴
ارائه چند برنامه مفید ۳۰۵
فصل ۱۰. آرایه‌ها و رشته‌ها ۳۳۱
۱۰-۱. آرایه‌های یک بعدی ۳۳۱
۱۰-۲. مرتب‌سازی آرایه‌ها ۳۳۷
۱۰-۳. جست‌وجو در آرایه‌ها ۳۴۳
۱۰-۴. رشته‌ها ۳۴۷
۱۰-۵. آرایه‌های چندبعدی ۳۵۵
۱۰-۶. مقداردهی آرایه‌ها ۳۶۶
۱۰-۷. آرایه‌ای از رشته‌ها ۳۶۹
تمرین ۳۷۴
ارائه چند برنامه مفید ۳۷۷
فصل ۱۱. اشاره‌گرها ۳۹۱
۱۱-۶. مبانی اشاره‌گرها ۳۹۱

۱۱-۲. اشاره‌گرها و نوع داده مبنا ۳۹۵
۱۱-۳. اشاره‌گرهای *void ۳۹۸
۱۱-۴. اَعمال روی اشاره‌گرها ۳۹۸
۱۱-۵. عملگرهای new و delete در متغیرهای پویا ۴۰۲
۱۱-۶. اشاره‌گرها و آرایه‌ها ۴۰۴
۱۱-۷. اشاره‌گرها و توابع (فراخوانی با ارجاع) ۴۰۷
۱۱-۸. اشاره‌گرهایی به توابع ۴۱۲
۱۱-۹. اشاره‌گرهایی به ثابت‌های رشته‌ای ۴۱۷
۱۱-۱۰. اشاره‌گر به اشاره‌گر ۴۲۱
۱۱-۱۱. اشتباهات متداول در به‌کارگیری اشاره‌گرها ۴۲۲
تمرین ۴۲۳
ارائه چند برنامه مفید ۴۲۶
واژه‌نامه انگلیسی به فارسی ۴۳۵
واژه‌نامه فارسی به انگلیسی ۴۴۵
منابع ۴۵۵

۱ مفاهیم الگوریتم و فلوچارت

۱-۱. مفهوم الگوریتم

واژه الگوریتم از نام ابوجعفر محمدبن‌موسی الخوارزمی ریاضی‌دان ایرانی در قرن دوم هجری شمسی گرفته شده است. این ریاضی‌دان برجسته ایرانی نگرش و اندیشیدن به راه حل مسئله یا طرز انجام عمل به جای عمل را مطرح نموده است که آن را امروزه الگوریتم می‌نامند. اگر فکر را به عنوان یک رویداد در نظر بگیریم و به جای واقعه فکر کردن به عمل فکر کردن بیندیشیم، روند انجام عمل الگوریتم فکری ما را تشکیل می‌دهد. بعد از خوارزمی هیچ کار درستی روی الگوریتم انجام نگردید، تا سال ۱۹۳۰ میلادی که روی توابع بازگشتی یا خودفراخوان کار شد و اولین مدل دقیق ریاضی برای مفهوم الگوریتم بعد از خوارزمی ارائه گردید.

همه ما به طور روزمره چند بار الگوریتم‌های مختلفی را اجرا می‌کنیم بدون آنکه آنها را به این نام بشناسیم. مثلاً چایی دم کردن، غذا پختن، خرید کردن و کار روزمره را انجام دادن هر کدام به شکل الگوریتم انجام می‌گیرند. همچنین وقتی روش پختن یک نوع شیرینی آموزش داده می‌شود، این پختن شیرینی طی مراحلی انجام می‌گیرد یعنی مجموعه‌ای از دستورالعمل‌ها آموزش داده می‌شود که باید اجرا گردد تا شیرینی پخته شود، یعنی انسان به عنوان مجری الگوریتم مجموعه‌ای از دستورالعمل‌ها را به مرحله اجرا می‌گذارد تا سرانجام شیرینی آماده شود [۱۷].

مجموعه دستورالعمل‌هایی که مراحل مختلف انجام کار یا راه حل مسئله‌ای را به زبان دقیق و با جزئیات کافی بیان نمایند، به نحوی که ترتیب توالی مراحل انجام آن و شرط خاتمه عملیات در آن کاملاً روشن و مشخص باشد، الگوریتم نامیده می‌شود. به عنوان مثال منطق ارائه شده در هر یک از مسئله‌های زیر نمونه‌ای از الگوریتم می‌باشد.

مثال ۱-۱: با نگرش به این که حجم کره از رابطه $V = \frac{۴\pi R^۳}{۳}$ و مساحت آن از رابطه $S = ۴\pi R^۲$ محاسبه می‌شود، منطق برنامه‌ای را طراحی نمایید که با اجرای آن شعاع کره به عنوان ورودی دریافت گردد، سپس مساحت و حجم کره محاسبه و چاپ شود.

الگوریتم مثال ۱-۱:

۱. عدد ۳٫۱۴۱۵۹ را در Pi قرار بده.
۲. عدد R را از ورودی دریافت کن.
۳. حاصل عبارت $۴Pi \times R^۲$ را در S قرار بده.
۴. حاصل عبارت $\frac{S \times R}{۳}$ را در V قرار بده.
۵. مقادیر S و V را چاپ کن و به اجرای الگوریتم پایان بده.

مثال ۱-۲: با نگرش به اینکه مساحت مثلث از رابطه

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

به دست می‌آید که در آن a، b و c طول اضلاع مثلث و p نصف مقدار محیط آن می‌باشد، منطق برنامه‌ای را طراحی نمایید که با اجرای آن مقادیر سه ضلع مثلثی از ورودی دریافت گردد، سپس محیط و مساحت مثلث محاسبه و چاپ شوند.

الگوریتم مثال ۱-۲:

۱. مقادیر a، b و c را از ورودی دریافت کن.
۲. مقادیر a، b و c را با هم جمع کن و بر ۲ تقسیم کن و حاصل تقسیم را در p قرار بده.
۳. مقدار حاصل از رابطه $\sqrt{p(p-a)(p-b)(p-c)}$ را در S قرار بده.
۴. مقادیر ۲p و S را چاپ کن و به اجرای الگوریتم پایان بده.

مثال ۱-۳: منطق برنامه‌ای را طراحی نمایید که با اجرای آن ضرایب a، b و c معادله درجه دوم $ax^۲ + bx + c = ۰$ از ورودی دریافت گردد، سپس ریشه‌های معادله مذکور محاسبه و چاپ شود. در صورت نداشتن ریشه حقیقی پیامی مبنی بر عدم وجود ریشه حقیقی چاپ گردد.

الگوریتم مثال ۱-۳:

۱. مقادیر a، b و c را از ورودی دریافت کن.
۲. دستورالعمل $D = b^2 - 4 \times a \times c$ را اجرا کن.
۳. اگر $D > 0$ به گام ۴ برو، در غیر این صورت به گام ۶ برو.
۴. دستورالعمل $x_1 = \frac{-b + \sqrt{D}}{2a}$ و $x_2 = \frac{-b - \sqrt{D}}{2a}$ را اجرا کن.
۵. مقادیر x_1 و x_2 را چاپ کن و به گام ۱۰ برو.
۶. اگر $D = 0$ به گام ۷ برو، درغیر این صورت به گام ۹ برو.
۷. دستورالعمل $x_1 = \frac{-b}{2a}$ را اجرا کن.
۸. مقادیر x_1 را که ریشه مضاعف است چاپ کن و به گام ۱۰ برو.
۹. پیامی به صورت "ریشه حقیقی ندارد" را چاپ کن.
۱۰. به اجرای الگوریتم پایان بده.

۱-۲. ویژگی‌های یک الگوریتم

هر الگوریتم باید دارای ۵ ویژگی زیر باشد (رجوع به [۱]، [۲] و [۷]):

۱) **متناهی بودن:** هر الگوریتم باید همیشه بعد از تعداد متناهی (محدود) مرحله خاتمه پذیرد.
۲) **صریح و روشن بودن:** هر مرحله از الگوریتم باید به طور دقیق تعریف شود و اَعمالی که اجرا می‌شوند باید روشن باشند.
۳) **مشخص بودن ورودی‌ها:** ورودی یک الگوریتم مقداری است که در ابتدا قبل از شروع داده شده است. ورودی یک الگوریتم یا صفر است یا یک یا بیشتر می‌باشد و این ورودی‌ها از مجموعه‌های مشخصی انتخاب می‌شوند. مثلاً در الگوریتم مثال ۱-۲ سه عدد قرار داده شده در a، b و c ورودی‌های الگوریتم هستند که از بین مجموعه اعداد صحیح مثبت انتخاب گردیده‌اند.
۴) **مشخص بودن خروجی‌ها:** هر الگوریتم یک یا چند خروجی دارد، مقادیری که رابطه مشخصی با ورودی‌ها دارند، خروجی‌ها را تشکیل می دهند. مثلا الگوریتم

مثال ۱-۲، دو خروجی به نام ۲p و S در گام ۴ دارد که به ترتیب محیط و مساحت مثلث هستند.

۵) **مؤثر بودن:** یک الگوریتم باید مؤثر باشد. این به آن معنی است که تمام عملیاتی که لازم است در الگوریتم انجام گیرند باید به حد کافی اساسی باشند، به نحوی که در اصل بتوانند در مدت زمان معینی توسط فردی با استفاده از مداد و کاغذ عیناً محاسبه گردند.

مثال ۱-۴: الگوریتم برنامه‌ای را طراحی نمایید که جمع اعداد فرد کوچک‌تر از ۱۰۰ را محاسبه و چاپ نماید.

الگوریتم مثال ۱-۴:

۱. یک جمع‌کننده به نام S انتخاب کن و عدد صفر را در آن قرار بده.
۲. یک شمارنده به نام C را انتخاب کن و عدد یک را در آن قرار بده.
۳. مقدار C را با مقدار S جمع کن و در S قرار بده.
۴. مقدار C را با ۲ جمع کن و در C قرار بده.
۵. اگر مقدار C کوچک‌تر از ۱۰۰ می باشد به گام ۳ برگرد.
۶. مقدار S را چاپ کن و به اجرای الگوریتم پایان بده.

مثال ۱-۵: الگوریتم برنامه‌ای را طراحی نمایید که سه عدد را از ورودی بخواند، سپس آنها را به صورت صعودی (الفبایی) مرتب و چاپ کند.

الگوریتم مثال ۱-۵:

۱. سه عدد را از ورودی دریافت کن و آنها را در A ، B و C قرار بده.
۲. اگر $A > B$ باشد، به گام ۳ برو، در غیر این صورت به گام ۶ برو.
۳. مقدار A را به X اختصاص بده. ($X \leftarrow A$)
۴. مقدار B را به A اختصاص بده. ($A \leftarrow B$)
۵. مقدار X را به B اختصاص بده. ($B \leftarrow X$)
۶. اگر $B > C$ باشد، به گام ۷ برو، در غیر این صورت به گام ۱۰ برو.
۷. مقدار B را به X اختصاص بده. ($X \leftarrow B$)
۸. مقدار C را به B اختصاص بده. ($B \leftarrow C$)

۹. مقدار X را به C اختصاص بده.($C \leftarrow X$)
۱۰. اگر $A > B$ باشد، به گام ۱۱ برو، در غیر این صورت به گام ۱۴ برو.
۱۱. مقدار A را به X اختصاص بده. ($X \leftarrow A$)
۱۲. مقدار B را به A اختصاص بده.($A \leftarrow B$)
۱۳. مقدار X را به B اختصاص بده.($B \leftarrow X$)
۱۴. محتوای A ، B و C را چاپ کن و به اجرای الگوریتم پایان بده.

۱-۳. زیرالگوریتم

برای به دست آوردن میانگین N عدد، اعداد را با هم جمع می‌کنیم و حاصل جمع را به N تقسیم می‌کنیم لذا هر زمان به محاسبه میانگین اعداد نیاز باشد، جمع اعداد لازم است. بدین لحاظ یک الگوریتم جهت محاسبه اعداد، می‌نویسیم و آن را $SUM(X, N, T)$ می‌نامیم که در آن X نام آرایه‌ای است که اعداد در آن ذخیره شده‌اند و N تعداد اعداد و T جمع اعداد است. و چون در مراحل اخیر الگوریتم‌های دیگر این الگوریتم فراخوانده می‌شود آن را زیر الگوریتم $SUM(X, N, T)$ می‌نامیم. در هنگام استفاده از زیرالگوریتم‌ها باید توجه نمود که تعداد شناسه‌های داده شده مساوی با تعدادی که در تعریف زیرالگوریتم مربوطه مشخص شده‌اند، باشد.

زیر الگوریتم $SUM(X, N, T)$:

۱. یک جمع کننده به نام T را انتخاب کن و عدد صفر را در آن قرار بده ($T = ۰$).
۲. یک شمارنده اعداد به نام I را انتخاب کن و عدد یک را در آن قرار بده ($I = ۱$).
۳. دستورالعمل $T = T + X(I)$ را اجرا کن.
۴. مقدار I را با یک جمع کن و در I قرار بده ($I = I + ۱$).
۵. اگر $I \leq N$ باشد به گام ۳ برگرد.
۶. مقدار T را چاپ کن و به الگوریتم فراخوان بازگردان.

۱-۴. فلوچارت یا رَوَندنما

فلوچارت یا رَوَندنما جهت نشان دادن داده‌ها، اطلاعات و جریان کار با استفاده از نمادهای خاصی که توسط خط جریان به هم پیوند داده می‌شوند، به کار برده می‌شود. با مروری بر روندنمای برنامه یا سیستم روند اجرای عملیات و ترتیب توالی مراحل اجرایی برنامه یا سیستم برای ما مشخص می‌گردد. ترکیب نمادها و خطوط جریان منطق برنامه یا سیستم را مجسم می‌سازد.

فلوچارت یک شکل خاص و راحت جهت بیان منطق برنامه می‌باشد و همانند نقشه شهر یا جاده می‌باشد که می‌تواند بهترین مسیر را جهت رسیدن به مقصد نشان دهد. رسم فلوچارت محدود به برنامه در زبان خاصی نمی‌باشد و جهت حل هر مسئله یا انجام کاری قبل از نوشتن برنامه باید فلوچارت‌های آن رسم گردد. اگر برنامهٔ نوشته‌شده آن طور که باید اجرا نشود، خطاها با استفاده از فلوچارت آسان‌تر کشف می‌شوند. جهت رسم یک فلوچارت به‌صورت دقیق و موفقیت‌آمیز، لازم است که برنامه‌نویس از ترتیب توالی مراحل مورد نیاز جهت به دست آوردن یک راه حل صحیح برای مسئله آگاهی کامل داشته باشد.

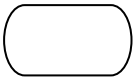
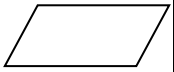
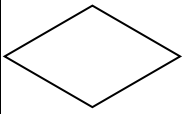
فلوچارت نه تنها یک نمایش تصویری از منطق برنامه را که برای برنامه‌نویس مفید است ارائه می‌دهد، بلکه جزء با ارزشی از مستندات برنامه می‌باشد که با کمک آن تفسیر برنامه و استفاده از برنامه با حداقل مشکلات توسط شخص دیگر امکان‌پذیر می‌باشد. فلوچارت بیانگر جریان کنترل بین دستورالعمل‌های مختلف قابل اجرا که برنامه را تشکیل می‌دهند، می‌باشد.

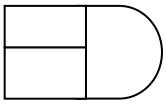
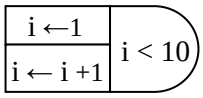
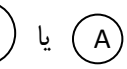
۱-۵. نمادهای مورد استفاده در فلوچارت

زبان مورد استفاده در فلوچارت‌ها شامل علامت کاملاً آشنای پیکان (فلش) و بعضی از نمادهای خاص که در این فصل شرح داده می‌شوند و عبارات و جملات و فرمول‌های ریاضی می‌باشد. پیکان‌ها جهت نشان دادن مسیر از یک مرحله به مرحله دیگر می‌باشند. نمادهای خاص هرکدام دارای یک شکل هندسی مخصوص می‌باشند که هدف ویژه نماد را مشخص می‌نماید.

عبارات و جملات و فرمول‌های ریاضی نوشته شده در اشکال هندسی نوع عملیاتی را که باید اجرا شود تجویز می‌نمایند. به طور متداول جریان عملیات که در روی فلوچارت نشان داده می‌شود از چپ به راست و از بالا به پایین نظیر نوشتار به زبان انگلیسی می‌باشد، با این

وجود پیکان‌ها جهت جریان را به طور روشن نشان می‌دهند، بدین لحاظ به پیکان‌ها خطوط جریان گفته می‌شود که ترتیب مراحل مختلف منطق برنامه را مشخص می‌سازند.
زمانی که از یک فلوچارت جهت نمایش تصویری الگوریتم استفاده می‌گردد، مراحل الگوریتم باید طبقه‌بندی گردند. طبقه‌بندی عمده مراحل به صورت ورودی، خروجی، تصمیم‌گیری و پردازش می‌باشد. در این فصل نمادهای فلوچارت مربوط به مراحل مذکور مورد بحث قرار می‌گیرند.

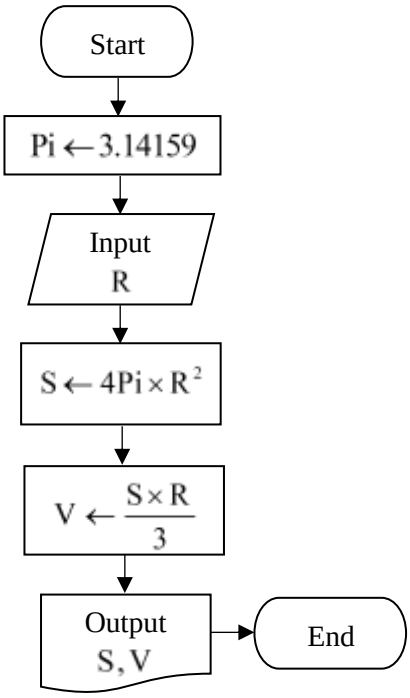
شکل هندسی	مفهوم شکل	مثال	مفهوم مثال
	شروع و پایان الگوریتم	Start	اجرای الگوریتم را شروع کن
	محاسبه و پردازش، جایگزینی	$S \leftarrow 0$ $p \leftarrow \frac{a+b+c}{2}$	مقدار ۰ را در S و مقدار $\frac{a+b+c}{۲}$ را در p قرار بده.
	ورود یا خواندن داده‌ها (از طریق صفحه کلید)	Input a, b, c	مقادیر a, b, c را وارد کن.
	خروج یا چاپ مقادیر و پیغام	Output S, V	مقادیر S و V را چاپ کن.
	تصمیم‌گیری	D > 0 T F	اگر D>۰ مسیر T و گرنه مسیر F را ادامه بده.
	فراخوان زیر الگوریتم‌ها	F(n)	زیرالگوریتم F را با آرگومان n فراخوانی کن.
	توضیحات جانبی	i عدد صحیح	عدد i در الگوریتم از نوع صحیح است.

	حلقه تکرار for		i شمارنده حلقه با مقدار اولیه ۱ و گام پیشرفت ۱. i<۱۰ شرط اجرای دستورات حلقه
	پیوند برنامه در یک صفحه		ادامه فلوچارت از ۱ یا از A پیگیری شود
	پیوند صفحات به یکدیگر		ادامه فلوچارت در صفحه ۲ پیگیری شود

۱-۶. رسم فلوچارت‌های نمونه

اینکه شما با فلوچارت و نمادهای تشکیل‌دهنده آن آشنا گردیده‌اید، در این فصل فلوچارت‌های متناظر با الگوریتم مثال‌های ۱-۱، ۱-۲، ۱-۳، ۱-۴ و ۱-۵ ارائه شده در بخش قبل ارائه می‌گردد، سپس در فصل‌های بعدی کتاب با طرح مسئله‌هایی الگوریتم‌ها و فلوچارت‌های نشانگر منطق برنامه هر مسئله ارائه می‌گردند.

فلوچارت مثال ۱-۱:



فلوچارت مثال ۱-۲: