مبانی رایانه و
برنامه‌نویسی به زبان
C++

تالیف:

مهندس رمضان عباس‌نژادورزی– مهندس فاطمه عبدی سقاواز
محمدعلی شمع‌علیزاده بائی – مهندس باقر رحیم‌پورکامی

فن‌آوری نوین

عنوان و نام پدیدآور	: مبانی ریانه و برنامه‌نویسی به زبان ++C/تالیف رمضان عباس‌نژادورزی... و[دیگران].
مشخصات نشر	: بابل:فن‌آوری نوین، ۱۳۹۳
مشخصات ظاهری	: :مصور، ۳۰۴ص.
شابک	: ۱۷۵۰۰۰ ریال:۵-۰۱-۷۲۷۲-۶۰۰-۹۷۸
یاداشت	: تالیف:رمضان عباس‌نژادورزی،فاطمه عبدی سقاواز، محمدعلی شمعلی زاده، باقر رحیم پور کامی
موضوع	: سی ++(زبان برنامه نویسی کامپیوتر)
موضوع	: الگوریتم‌های کامپیوتری
موضوع	: زبان برنامه نویسی کامپیوتر
شناسه افزوده	: عباس‌نژاد ورزی، رمضان، ۱۳۴۸ -
رده بندی کنگره	: ۲۳۱۳۹۲ م ۹۳ س/۷۶/۷۳ QA
رده بندی دیویی	: ۰۰۵ / ۷۳
شماره کتابشناسی ملی	: ۳۳۴۴۳۴۱

فن‌آوری نوین بابل، کدپستی ۴۷۱۶۷-۷۳۴۴۸ www.fanavarienovin.net تلفن: ۲۲۵۶۶۸۷-۰۱۱۱

مبانی رایانه و برنامه‌نویسی به زبان C++

تألیف: مهندس رمضان عباس‌نژادورزیی – مهندس فاطمه عبدی سقاواز- محمد علی شمعلی‌زاده- مهندس باقر رحیم‌پور کامی

نوبت چاپ: چاپ دوم

سال چاپ: زمستان ۱۳۹۳

شمارگان:۱۰۰۰ جلد

قیمت: ۱۷۵۰۰ تومان

نام چاپخانه و صحافی: فرنگاررنگ

شابک: ۵-۰۱-۷۲۷۲-۶۰۰-۹۷۸

نشانی ناشر: بابل، چهارراه نواب، کاظم‌بیگی، جنب حسینیه منصور کاظم‌بیگی، طبقه همکف

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۲۲۰-۶۶۴۰۰۱۴۴

فهرست مطالب

فصل اول: آشنایی با رایانه، الگوریتم و فلوچارت

۱- ۱. انواع رایانه‌ها.....۷

۲- ۱. سخت‌افزار و نرم‌افزار.....۸

۳ – ۱. پردازنده مرکزی.....۸

۴ – ۱. اجزای تشکیل‌دهنده رایانه.....۱۰

۱ – ۴– ۱. واحد ورودی.....۱۰

۵ – ۱. واحد حافظه.....۱۳

۱ – ۵ – ۱. حافظه اصلی.....۱۴

۲–۵ – ۱. حافظه کمکی.....۱۴

۳–۵ – ۱. واحدهای اطلاعاتی و حافظه.....۱۷

۶ – ۱. واحد خروجی.....۱۸

۷ – ۱. انتقال اطلاعات و پیدایش شبکه.....۲۰

۸ – ۱. مراحل برنامه‌نویسی.....۲۰

۹- ۱. الگوریتم چیست؟.....۲۱

۱۰ – ۱. تفکر الگوریتمیک.....۲۲

۱۱ – ۱. تعریف فلوچارت.....۲۳

۱۲ – ۱. ساختار تصمیم.....۲۵

۱۳ – ۱. ساختار حلقه.....۲۶

۱۴ – ۱. آرایه.....۴۴

۱۵ – ۱. الگوریتم فرعی.....۴۶

۱۶– ۱. مسائل حل شده.....۴۸

۱۷–۱. سیستم اعداد دودویی.....۵۷

۱۸ – ۱. تمرین‌ها.....۶۱

فصل دوم: آشنایی با زبان C++

۱ – ۲. سطوح مختلف زبان‌های برنامه سازی.....۶۵

۱ – ۱ – ۲. زبان‌های سطح پایین.....۶۵

۲ – ۱ – ۲. زبان‌های سطح بالا.....۶۶

۳ – ۱ – ۲. زبان‌های سطح میانی.....۶۷

۲ – ۲. ویژگی‌های زبان برنامه‌نویسی C++.....۶۷

۳ – ۲. آموزش زبان‌های برنامه‌نویسی.....۶۸

۴ – ۲. کلمات کلیدی.....۶۸

۵- ۲. انواع داده‌ها.....۶۹

۱ – ۵ – ۲. داده‌های اولیه.....۷۰

۶ – ۲. متغیر.....۷۲

۷ – ۲. ثابت‌ها.....۷۲

۸ – ۲. عملگرها.....۷۲

۱ – ۸ – ۲. عملگرهای محاسباتی.....۷۳

۲ – ۸ – ۲. عملگرهای رابطه‌ای (مقایسه‌ای).....۷۴

۳ – ۸– ۲. عملگرهای ترکیبی.....۷۴

۴ – ۸ – ۲. عملگرهای منطقی.....۷۴

۵ – ۸ – ۲. عملگرهای خاص.....۷۵

۹ – ۲. اولویت عملگرها.....۷۷

۱۰ – ۲. تبدیل نوع.....۷۸

۱۱ – ۲. ساختار برنامه C++.....۷۹

۱۲ – ۲. دستورات ورودی و خروجی.....۸۱

۱ – ۱۲ – ۲. دستورات ورودی.....۸۱

۲ – ۱۲ – ۲. دستورات خروجی.....۸۱

۱۳ – ۲. گرافیکی در C++.....۸۵

۱۴ – ۲. مسائل حل شده.....۹۰

۱۵ – ۲. مسائل حل شده در سایت.....۹۲

۱۶ – ۲. تمرین‌ها.....۹۳

فصل سوم: ساختار تصمیم و تکرار

۱ – ۳. ساختارهای تصمیم‌گیری.....۹۷

۱ – ۱ – ۳. ساختار تصمیم if.....۹۷

۲ – ۱ -۳. ساختار if تو در تو.....۹۹

۳ – ۱ – ۳. ساختار switch ۱۰۳
۲ – ۳. ساختارهای تکرار ۱۰۴
۱ – ۲ – ۳. ساختار تکرار for ۱۰۵
۲ – ۲ – ۳. دستور break ۱۰۶
۳ – ۲ – ۳. دستور continue ۱۰۶
۴ – ۲ – ۳. ساختار while ۱۰۹
۵ – ۲ -۳. ساختار تکرار do while ۱۱۱
۳ – ۳. مسائل حل شده ۱۱۴
۴ – ۳. مسائل حل شده در سایت ۱۲۶
۵ – ۳. تمرین‌ها ۱۲۹

فصل چهارم: توابع

۱ – ۴. انواع توابع ۱۳۶
۲ – ۴. توابعی که برنامه نویس می‌نویسد ۱۳۶
۱ – ۲ – ۴. نوشتن تابع ۱۳۶
۲ – ۲ – ۴. فراخوانی تابع ۱۳۶
۳ – ۴. ارسال پارامترها ۱۴۴
۱ – ۳ -۴. ارسال پارامتر از طریق مقدار ۱۴۴
۲ – ۳ – ۴. ارسال پارامتر از طریق ارجاع ۱۴۴
۴ – ۴. طول عمر و محدوده حضور متغیرها ۱۴۶
۱ – ۴ – ۴. طول عمر متغیر ۱۴۶
۲ – ۴ – ۴. محدوده حضور متغیر ۱۴۷
۵ – ۴. ارسال پارامتر از طریق ارجاع ۱۵۰
۶ – ۴. توابع inline ۱۵۳
۷ – ۴. چند ریختی توابع ۱۵۴
۸ – ۴. تعریف آرگومان‌های اختیاری با مقدار پیش‌فرض ۱۵۵
۹ – ۴. توابع بازگشتی ۱۵۶
۱۰ – ۴. معرفی چند تابع کتابخانه‌ای ۱۵۹
۱۱ – ۴. مسائل حل شده ۱۵۹
۱۲ – ۴. مسائل حل شده در سایت ۱۷۱
۱۳– ۴. تمرین‌ها ۱۷۵

فصل پنجم: آرایه‌ها، رشته‌ها و اشاره‌گرها

۱ – ۵. آرایه‌های یک بعدی ۱۸۰
۲- ۵. مقداردهی به عناصر آرایه ۱۸۱
۱ – ۲ – ۵. مقداردهی به عناصر آرایه به صورت خانه‌های مجزا ۱۸۱
۲ – ۲ – ۵. مقداردهی اولیه به آرایه در هنگام تعریف آن ۱۸۱
۳ – ۲ -۵. مقداردهی به عناصر آرایه با حلقه تکرار و شیء cin ۱۸۲
۳ – ۵. نمایش عناصر آرایه ۱۸۳
۱ – ۳ -۵. نمایش مقادیر هر عنصر به صورت مجزا ۱۸۳
۲ – ۳ – ۵. نمایش عناصر آرایه با حلقه تکرار ... ۱۸۳
۴ – ۵. تولید اعداد تصادفی ۱۸۴
۵ – ۵. مرتب سازی حبابی ۱۸۵
۶ – ۵. جست‌وجوی مقدار در آرایه ۱۸۶
۱ – ۶ – ۵. جست‌وجوی خطی (ترتیبی) ۱۸۶
۲ – ۶ – ۵. جست‌وجوی دودویی ۱۸۷
۷ – ۵. آرایه‌های دو بعدی ۱۹۵
۸ – ۵. تعریف آرایه دو بعدی ۱۹۵
۹ – ۵. مقداردهی به عناصر آرایه دو بعدی ۱۹۶
۱ – ۹ – ۵. مقداردهی اولیه عناصر آرایه دو بعدی ۱۹۶
۲ – ۹ -۵. مقداردهی به عناصر آرایه دو بعدی با حلقه‌های تودرتو و شیء cin ۱۹۶
۱۰ – ۵. نمایش مقادیر عناصر آرایه دوبعدی ۱۹۷

۱۱ – ۵. رشته‌ها..... ۲۰۲
۱ – ۱۱ – ۵. مقداردهی به رشته‌ها..... ۲۰۲
۱۲ – ۵. توابع رشته‌ای..... ۲۰۶
۱۳ – ۵. تعریف آرایه‌ای از رشته‌ها ۲۱۰
۱ – ۱۳ – ۵. مقداردهی به آرایه‌ای از رشته‌ها..... ۲۱۰
۲ – ۱۳–۵. نمایش محتویات آرایه رشته‌ای ۲۱۱
۱۴ – ۵. اشاره‌گرها..... ۲۱۳
۱ – ۱۴ – ۵. توابع و اشاره‌گرها..... ۲۱۴
۲ – ۱۴ – ۵. اشاره‌گرها و آرایه‌ها..... ۲۱۶
۱۵–۵. تخصیص پویای حافظه..... ۲۱۸
۱۶ – ۵. مسائل حل شده..... ۲۲۰
۱۷ – ۵. مسائل حل شده در سایت..... ۲۳۸
۱۸ – ۵. تمرین‌ها..... ۲۴۱
پیوست الف: مسائل تکمیلی و امتحانی..... ۲۴۷
پیوست ب: پروژه برنامه نویسی..... ۲۸۷
منابع: ۳۰۴

کتاب‌های منتشر شده انتشارات فن‌آوری نوین	
نام کتاب	**انتشارات**
حل مسائل C	فن آوری نوین
حل مسائل C++	فن آوری نوین
آموزش گام به گام برنامه نویسی بانک اطلاعات با C#	فن آوری نوین
حل مسائل C#	فن آوری نوین
حل مسائل پاسکال	فن آوری نوین
آموزش گام به گام برنامه نویسی بانک اطلاعات با ویژوال‌بیسیک نت	فن آوری نوین
آموزش گام به گام LINQ با C#	فن آوری نوین
تجارت الکترونیکی	فن آوری نوین
آشنایی با مبانی امنیت شبکه	فن آوری نوین
کاربرد رایانه در مدیریت و حسابداری	فن آوری نوین
مدیریت استراتژیک فن‌آوری اطلاعات	فن آوری نوین
طراحی سیستم‌های شی‌گرا با زبان C#	فن آوری نوین
اصول طراحی پایگاه داده	فن آوری نوین
آموزش گام به گام برنامه نویسی به زبان C++	فن آوری نوین
گرافیک رایانه‌ای با زبان C#	فن آوری نوین
آزمایشگاه پایگاه داده با SQL Server 2012	فن آوری نوین
سنجش از دور کاربردی (جلد اول)	فن آوری نوین
ساختمان داده‌ها با C++	فن آوری نوین
فیزیک الکتریسیته	فن آوری نوین

مقدمه

گسترش و نقش آفرینی نرم‌افزارهای رایانه‌ای و استفاده روز افزون آن‌ها در بخش‌های مختلف نظیر صنعت، تجارت، پزشکی، علوم، اداری، حسابداری و غیره، روز به روز بر اهمیت یادگیری زبان مناسب برنامه‌نویسی می‌افزاید.

زبان‌های برنامه‌نویسی زیادی وجود دارند. در بین این زبان‌ها C++ از ویژگی‌های بخصوصی از لحاظ آموزشی و کاربردی برخودار است. به طوری که این زبان یکی از جذاب‌ترین و قدرت‌مندترین زبان‌های برنامه‌نویسی **شیءگرا** است. از طرف دیگر، زبان C++ به عنوان سرفصل درس مبانی برنامه‌سازی و برنامه‌سازی پیشرفته در رشته-های کامپیوتر، فناوری اطلاعات، ICT، علوم کامپیوتر و رشته‌های فنی و مهندسی تدریس می‌شود.

کتاب‌های زیادی در زمینه برنامه‌نویسی C++ نوشته شده‌اند که جای تقدیر و تشکر دارد. اما، کتاب حاضر با سال-ها تجربه در زمینه تالیف کتب برنامه‌نویسی و تدریس این زبان تدوین شده است و دارای ویژگی‌های زیر است:

- **بیان الگوریتم و فلوچارت** که پایه و اساس یادگیری هر زبان برنامه‌نویسی است.
- **ارائه و حل برنامه‌های متعدد،** در این کتاب حدود 506 مسئله برنامه‌نویسی بیان گردیده، که 180 برنامه آن در متن کتاب حل شده است. حل حدود 142 برنامه در سایت انتشارات فن‌آوری نوین آمده است. اما، 184 برنامه بدون حل باقی مانده است که به عهده دانشجویان محترم می‌باشد.
- در پیوست الف کتاب **حدود ۵۰ سوال تکمیلی و امتحانی** با توضیحات کامل حل گردیده است.
- در پیوست ب کتاب **پروژه مین روب** به عنوان یک پروژه عملی با توضیحات به طور کامل پیاده‌سازی شده است.
- **سادگی و روانی کتاب،** کتاب به صورت گام‌به‌گام با جملات کوتاه و ساده بیان گردیده است.

امیدواریم این اثر نیز مورد توجه اساتید و دانشجویان عزیز واقع شود.

مسائل حل شده در سایت که انتهای هر فصل کتاب آمده است را می‌توانید از آدرس سایت انتشارات فن-آوری نوین به آدرس www.fanavarienovin.net دریافت نمایید.

برنامه‌های کتاب در یکی از محیط‌های TC(Turbo C) و Borland C++ تحت ویندوز(این کامپایلر را می‌توانید در سیستم عامل ویندوز ۷ نیز نصب کنید) تست و اجرا شده‌اند که نمونه خروجی آن‌ها در کتاب آمده است. برنامه‌های که در آن‌ها محیط گرافیک فعال می‌شوند را باید با کامپایلر TC تست کرده و خروجی بگیرید.

در پایان از تمامی خوانندگان عزیز (اساتید و دانشجویان) تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهادات و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.

بابل، زمستان۱۳۹۳

مولفین

فصل ۱ آشنایی با رایانه، الگوریتم و فلوچارت

اولین قدم برای استفاده از هر ابزاری شناخت و بیان کاربردهای آن است. بنابراین، برای استفاده از رایانه باید آن را شناخت و کاربردهای آن را دانست. یعنی، ابتدا باید تعیین کرد رایانه چیست؟، چه ویژگی‌ها و کاربردهای دارد؟ یا مهم‌تر این که با چه دیدی باید به رایانه نگاه نمود تا سودمند باشد و ضرر نداشته باشد. بسیاری از اشخاص، ادارات، سازمان‌ها، حتی مهندسین، ... رایانه را به شکل فانتزی نگاه می‌کنند. آن‌ها انتظار دارند تنها با خرید یک رایانه، مشکلات سازمان و ادارات‌شان برطرف شود. این افراد رایانه را نشناختند و کاربردهای آن را نمی‌دانند. برای این که بتوان رایانه را شناخت، باید مشخص نمود چرا رایانه اختراع شد؟ چون به آن نیاز داشتند، آن را ساختند. پس، نیاز است که موجب اختراع تکنولوژی‌ها و ابزارهای جدید می‌شود. بنابراین، باید تعیین کرد که رایانه چه نیازهایی از بشر را برطرف می‌کند. پیچیدگی و سختی کارها، دقت و صحت، تکرار، حجم زیاد اطلاعات، انتقال اطلاعات، سرعت بالا و کاهش هزینه، مهم‌ترین عواملی هستند که نیاز به رایانه را بیان می‌کنند. **رایانه دستگاه الکترونیکی قابل برنامه‌ریزی است که توسط ابزارهای ورودی داده را از دنیای خارج دریافت نموده، توسط پردازش گر پردازش می‌نماید تا اطلاعات تولید گردد و اطلاعات را در حافظه نگهداری کرده، در پایان توسط ابزارهای خروجی اطلاعات را به دنیای خارج منتقل می‌کند.**

۱– ۱. انواع رایانه‌ها

برای آشنایی با انواع رایانه، فرض کنید بخواهید از نقطه‌ای به نقطه دیگر مسافرت کنید. برای انجام این کار وسائل نقلیه از قبیل ماشین، هواپیما، دوچرخه، قطار و غیره وجود دارند. اگر فاصله سفرتان طولانی نباشد، از ماشین استفاده خواهید کرد. ولی، اگر سفرتان طولانی باشد، از هواپیما استفاده می‌کنید. رایانه‌ها نیز کاربردهای مختلفی دارند. با توجه به انواع کاربردها، رایانه‌های متعددی ساخته شده‌اند. برخی از انواع رایانه‌ها عبارت‌اند از:

- **ابر رایانه‌ها**[1]، قدرتمندترین رایانه‌هایی هستند که تاکنون ساخته شده‌اند. در ساختمان این رایانه‌ها، تعداد زیادی پردازنده وجود دارند که با همکاری هم کار می‌کنند. در سال، تعداد اندکی از این رایانه تولید می‌شوند، چون سازمان‌های کمی در دنیا به چنین توانایی و قدرت پردازش نیاز دارند. از طرف دیگر، هزینه‌های تولید این رایانه‌ها بسیار بالا است. بنابراین، هر سازمانی نمی‌تواند چنین هزینه‌هایی را بپردازد.
- **رایانه‌های بزرگ**[2]، یک پردازنده قدرتمند دارند که به طور موازی چندین کاربر می‌توانند از آن استفاده کنند. این رایانه‌ها، دیگر تولید نمی‌شوند و به تدریج از رده خارج شدند. افزایش قدرت کارایی و کاهش قیمت رایانه‌های شخصی امروزی، این رایانه‌ها را غیر قابل استفاده کرده است.

---

[1] .Supper Computers [2].Main Frame Computers

رایانه‌های کوچک[1]، برای استفاده در سازمان‌ها، ادارات و شرکت‌های متوسط ساخته شده‌اند. این رایانه‌ها، اندازه‌های متعددی دارند. برخی از آن‌ها رومیزی هستند یا اندازه آن‌ها تا یک کابینت متغیر است.

رایانه‌های شخصی[2]، رایانه‌های امروزی هستند که به رایانه‌های رومیزی معروفند. امروزه، این رایانه‌ها در اکثر منازل، ادارات و سازمان‌ها وجود دارند و کاربردهای عمومی دارند. زیرا، نه تنها قیمت آن‌ها بسیار پایین است، بلکه ممکن است صدها برابر یک رایانه بزرگ قدرت داشته باشند.

رایانه‌های کیفی[3]، رایانه‌های قابل حمل هستند که می‌توانند با باطری و برق شهر کار کنند. ویژگی قابل حمل بودن این رایانه‌ها موجب شده است که انسان‌ها بتوانند در مسافرت، کارهای خودشان را انجام دهند. یعنی، کارمندان بدون این که به اداره محل خدمتشان مراجعه کنند، می‌توانند کارهایشان را انجام دهند. این روش انجام کار، **دور کاری** نام دارد.

رایانه‌های جیبی[4]، همان‌طور که از نام آن‌ها پیداست، اندازه آن‌ها کوچک و وزن آن‌ها نیز کم است. از طرف دیگر، در جیب جا می‌شوند. بنابراین، به سادگی قابل حمل هستند.

۲– ۱. سخت‌افزار و نرم‌افزار

هر فردی که با رایانه آشنایی دارد، واژه‌های سخت‌افزار و نرم‌افزار را شنیده است. برای درک بهتر این مفاهیم یک کلاس آموزش رایانه را در نظر بگیرید. در این کلاس، میز، صندلی، تخته، ویدئو پروژکتور، رایانه، استاد، دانشجو و ابزارهای دیگر وجود دارند. تمام این ابزارها و افرادی که در کلاس وجود دارند، سخت‌افزار کلاس درس می باشند. چون، قابل رویت و لمس هستند. بنابراین، تمام چیزهایی که قابل لمس و رویت باشند، **سخت‌افزار**[5] نام دارند. در رایانه، دستگاه‌هایی از قبیل صفحه کلید، صفحه نمایش، ماوس، دسته بازی، حافظه، CD، DVD، پردازنده مرکزی، کارت‌های گرافیکی و ابزارهای دیگر که قابل لمس و رویت هستند، سخت‌افزار رایانه می‌باشند. اکنون، برمی‌گردیم به مثال کلاس درس، همان‌طور که بیان گردید، تمام ابزارها و افرادی که در کلاس درس وجود دارند، سخت‌افزار هستند. حال، سوال این است که نرم‌افزار کلاس درس چیست؟ برای پاسخ به این سوال، فرض کنید کلاس درس و تمام امکانات آن آماده باشد، آیا شما می‌توانید آن کلاس را اداره کرده و آموزش دهید؟ آیا استاد دیگری که تخصص آن رایانه نباشد می‌تواند از امکانات کلاس استفاده کند و رایانه آموزش دهد؟ پاسخ به این سوالات منفی است. پس، نرم‌افزار کلاس درس، همان علم استاد رایانه است که ابزارهای فراهم شده در کلاس درس را مورد بهره برداری قرار می‌دهد. بنابراین، **نرم‌افزار رایانه**[6]، علمی است که سخت‌افزار رایانه را راه‌اندازی کرده، مورد بهره‌برداری قرار می‌دهد.

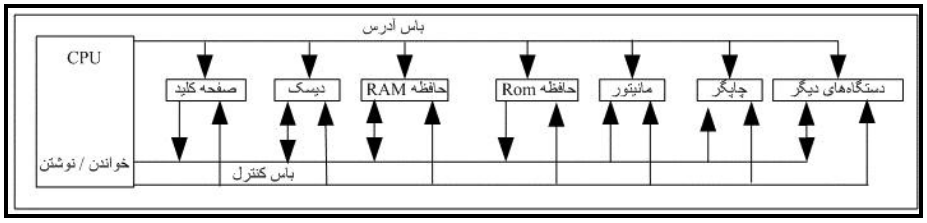
۳ – ۱. پردازنده مرکزی

پردازنده مرکزی، قسمتی از رایانه است که عملیات محاسباتی و منطقی را انجام می‌دهد. نام دیگر پردازنده، **واحد پردازشگر مرکزی (CPU)** است. CPU، یکی از قطعات بسیار مهم رایانه است که بر روی مادربرد

---

[1].Mini Computers [2]. Personal Computers [3] .Laptop Computers [4].Portable Computers
[5] .Hardware [6].Computer Software

(Mainboard) قرار می‌گیرد. وظیفه و عملکرد آن در رایانه، مشابه با مغز انسان است، همان‌گونه که مغز انسان وظیفه پردازش اطلاعات دریافتی و نظارت بر سایر اعضای بدن را به عهده دارد، یک CPU نیز پردازش داده‌ها را انجام می‌دهد و همچنین نظارت بر کار دیگر اجزای رایانه را به عهده دارد. ارتباط پردازنده با سایر اجزای رایانه از طریق باس‌ها (BUS) انجام می‌شود. هر پردازنده سه نوع باس دارد. این باس‌ها عبارت‌اند از (شکل ۱ – ۱).



شکل ۱ – ۱ ارتباط CPU با اجزای رایانه از طریق باس‌ها.

- **باس داده (Data Bus)**، برای انتقال داده بین اجزای رایانه به کار می‌رود.
- **باس آدرس (Address Bus)**، برای تعین آدرس حافظه مورد نیاز اجزای رایانه به کار می‌رود.
- **باس کنترل (Control Bus)**، برای کنترل و هدایت عملیات خواندن و نوشتن توسط اجزای رایانه به کار می‌رود.

یک پردازنده از سه بخش اصلی تشکیل شده است که عبارت‌اند از (شکل ۲ – ۱).

۱. واحد حافظه (MU)[1]، وظیفه آن نگهداری داده‌ها است. این واحد از ثبات‌ها[2]، حافظه RAM و ROM تشکیل شده است. ثبات‌ها، از چند عنصر الکترونیکی تشکیل می‌شوند. این عناصر به طور منطقی در مجاور هم قرار می‌گیرند. وظیفه ثبات‌ها نگهداری موقت داده و دستورالعمل‌ها است. رایانه‌های امروزی ثبات‌های ۸، ۱۶، ۳۲، ۶۴ بیتی دارند. با حافظه‌های RAM و ROM در ادامه آشنا خواهید شد.

۲. واحد محاسبه و منطق(ALU)[3]، وظیفه انجام عملیات محاسباتی و منطقی را بر عهده دارد. اعمالی از قبیل جمع، ضرب، تفریق و تقسیم، عملیات محاسباتی نام دارند. ولی، اعمالی از قبیل مقایسه دو مقدار، عملیات منطقی نامیده می‌شوند. یعنی، از وظایف این واحد، تصمیم‌گیری در اعمال منطقی (مقایسه‌ای) مثل بزرگ‌تر، مساوی، بزرگ‌تر مساوی و ... است. به عنوان مثال، دستورات زیر را در نظر بگیرید:

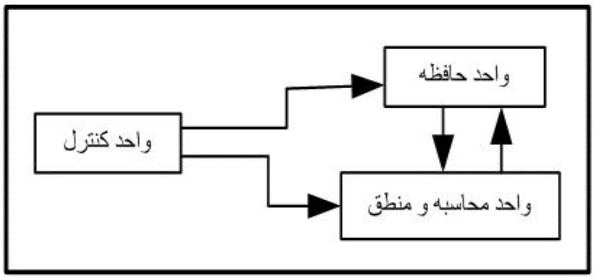
- اگر حقوق کوچک‌تر یا مساوی ۸۳۳۰۰۰۰ ریال است، مالیات برابر با صفر است. وگرنه، مالیات برابر با ۸۳۳۰۰۰۰ – حقوق) ضرب در ۰,۱ است.
- مالیات را چاپ کن.

این تصمیم‌گیری جهت محاسبه مالیات و چاپ آن به کار می‌رود.

- **واحد کنترل (CU)**[4]، مسئول دریافت دستورات، داده‌ها، رمزگشایی آن‌ها، سازماندهی و کنترل واحد محاسبه و منطق است. یعنی، بدون اجازه واحد کنترل، هیچ عملی انجام نمی‌شود. واحد کنترل اجازه خواندن دستورات، خواندن داده، نوشتن داده، انجام عملیات (اجرای دستورات) را صادر می‌کند.

---

[1] .Memory Unit [2].Registers [3].Arithmatic and Logical Unit [4].Control Unit



شکل ۲ – ۱ ساختار پردازشگر مرکزی (CPU).

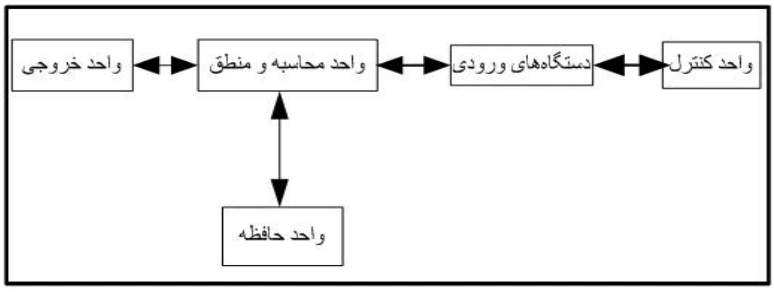
۴ – ۱. اجزای تشکیل‌دهنده رایانه

قبل از این که به اجزای تشکیل‌دهنده رایانه بپردازیم، فرض کنید، رایانه بخواهد حقوق کارمندان را دریافت و محاسبه کند. بنابراین، باید واحدی داشته باشد که بتواند احکام کارکنان را دریافت نماید. این **واحد ورودی**[1] نام دارد. علاوه بر دریافت احکام باید دارای واحدی باشد که بتواند این احکام را ذخیره نماید. این **واحد حافظه**[2] نام دارد. پس از انتقال احکام به واحد حافظه، رایانه باید بتواند قوانین محاسبه حقوق را بر روی احکام اعمال کرده، بیمه، مالیات، اضافه کار و غیره را محاسبه کند. انجام این اعمال وظیفه **واحد پردازش‌گر مرکزی (CPU)**[3] است. بعد از محاسبه حقوق، رایانه باید بتواند نتایج را به کارمندان تحویل دهد و لیست‌های متعددی را تهیه نماید. این کار را **واحد خروجی**[4] انجام می‌دهد. بنابراین، اجزای اصلی رایانه عبارت‌اند از:

۱. واحد ورودی ۲. واحد حافظه ۳. واحد محاسبه و منطق ۴. واحد خروجی (شکل ۳ – ۱).

۱ – ۴– ۱. واحد ورودی

واحد ورودی، وظیفه دریافت داده‌ها از دنیای خارج و انتقال آن به رایانه را دارد. از آنجایی که باید بتوان داده‌های مختلف(از قبیل متن، اعداد، تصاویر، امضاء، بارکد و غیره) را به رایانه انتقال داد، ابزارها و دستگاه‌های متعدد ورودی ساخته شده‌اند. برخی از این دستگاه‌ها را در ادامه می‌بینید.



شکل ۳ – ۱ اجزای تشکیل دهنده رایانه.

صفحه کلید[5]، رایج‌ترین دستگاه ورودی در رایانه است. صفحه کلید از تعدادی کلید از قبیل کلیدهای عددی، حرفی، کنترلی، تابعی (Fها)، جهت نماها، نشانه‌ها و غیره تشکیل شده است. برخی از این کلیدها در

---

[1].Input Unit [2]. Memory Unit [3].Central Processing Unit [4].Output Unit
[5] .Keyboard

تمام نرم‌افزارها یک کاربرد دارند (نظیر کلیدهای حرفی و عددی). اما، بعضی دیگر از کلیدها مانند کلیدهای تابعی و کنترلی در نرم‌افزارهای مختلف کاربردهای متفاوتی دارند. تعداد کلیدهای صفحه کلید متفاوت است (صفحه کلیدهای قدیمی ۱۰۱ کلید داشتند. اما، صفحه کلیدهای جدید ۱۰۴ یا بیشتر کلید دارند). صفحه کلید از مهم‌ترین دستگاه‌های ورودی یک برنامه نویس است.

- **ماوس**[1]، پس از صفحه کلید، پرکاربردترین دستگاه ورودی است. ماوس‌ها دو یا سه کلید دارند. ماوس، وظیفه انتقال مکان‌نما و انتخاب گزینه‌ها روی صفحه نمایش یا اجزای برنامه را دارد. یعنی، با حرکت آن روی یک سطح صاف، هم‌زمان مکان‌نما (اشاره‌گر ماوس) روی صفحه نمایش جابه‌جا شده و با فشار دکمه‌ای از ماوس بر روی آیکون یا برنامه کار خاصی انجام می‌شود. به عنوان مثال، اگر بر روی برنامه‌ای، دکمه سمت چپ ماوس را فشار دهید[2]، آن برنامه اجرا خواهد شد.
- **گوی‌های کنترل**، همانند ماوس است. با این تفاوت که گوی (در گوی کنترل) در قسمت بالای آن قرار گرفته و توسط حرکت دست در جهات مختلف چرخیده و مکان‌نمای آن روی صفحه نمایش جابه‌جا می‌شود.
- **صفحه نمایش‌های لمسی**[3]، به گونه‌ای طراحی و ساخته شده‌اند که به حرکت و فشار دست بر روی آن حساس هستند. بنابراین، مانند ماوس می‌توان با حرکت دست، برنامه‌ای را انتخاب کرده یا آن را اجرا نمود. برخی از موبایل‌ها نیز صفحه نمایش لمسی دارند.
- **قلم نوری**، یک قلم حساس به نور است که با کشیدن آن بر روی صفحه نمایش (مانند ماوس) می‌توان اعمالی از قبیل انتخاب و رسم اشکال را انجام داد.
- **جوی استیک**(دسته بازی)[4]، دارای دسته‌ای است که در بازی‌های رایانه‌ای برای انتقال مکان‌نما و اجرای دستورات به کار می‌رود (شکل ۴ – ۱).
- **صفحه ورود اطلاعات**، تخته‌ای است که به حرکت و فشار بر روی آن حساس است. به طوری که با حرکت بر روی این تخته و فشردن دست، تصویری (نقاشی) در رایانه رسم خواهد شد. این دستگاه برای ترسیم نقاشی، از ماوس کاراتر و آسان‌تر است.
- **کارت خوان‌ها**[5]، کارت‌های مغناطیسی یا کارت‌هایی که بارکد یا علامت خاصی بر روی آن‌ها ذخیره شده باشد (مانند کارت‌های ورود و خروج ادارات، بارکدهای چاپ شده بر روی کالاها، کارت‌های اعتباری، کارت‌های هوشمند) را می‌خوانند. کارت‌ها، مغناطیسی یا نوری هستند. برخی از کارت خوان‌های مغناطیسی به تماس مستقیم کارت به کارت خوان نیاز دارند. این کارت خوان‌ها، **تماسی** نام دارند. ولی، بعضی از آن‌ها نیاز به تماس کارت به کارت خوان ندارند (مانند کارت خوان‌های نصب شده در متروها، ایستگاه‌های قطار). این کارت خوان‌ها، **بدون تماس** نامیده می‌شوند. امروزه اکثر فروشگاه‌ها از کارت خوان استفاده می‌کنند.

---

[1].Mouse [2].Click [3].Touch Screens [4].Joy Stick [5].Card Readers

شکل ۴ – ۱ نمونه‌هایی از جوی استیک.

حسگرها[1]، دستگاه‌هایی هستند که با وقوع عمل خاص، واکنش معینی را نشان می‌دهند. امروزه، انواع مختلف حسگرها از قبیل حسگرهای گرمایی، نور، صدا و حرکت وجود دارند. به عنوان مثال، حسگر دما، درجه حرارت محیط را به رایانه انتقال می‌دهد. بنابراین، با یک حسگر دما، یک برنامه رایانه‌ای و یک کولر می‌توان دمای محیط اتاق را تنظیم کرد.

علامت خوان کاغذ[2]، برای خواندن علامت‌های خاص (نظیر برگه‌های کنکور) به کار می‌روند. این علامت خوان‌ها، برگه‌ها را به داخل خود کشیده و علامت ثبت شده روی آن‌ها را می‌خوانند. علامت خوان‌ها به کربن (کربن خوان‌ها) و یا نوشته‌ای با رنگ خاص (نظیر سیاه) حساس هستند.

دستکش‌های واقعیت مجازی، جهت انتقال حرکت دست و حتی شدت آن به رایانه به کار می‌روند. دست انسان‌ها در این دستکش‌ها قرار می‌گیرد.

دیجیتایزر، برای انتقال تصاویر، نقشه‌ها و طرح‌ها به رایانه به کار می‌روند و از دو قسمت قلم نوری و صفحه گرافیکی تشکیل شده‌اند. با کشیدن قلم نوری بر روی صفحه گرافیکی، تصویر به رایانه انتقال یافته بر روی صفحه نمایش، ظاهر می‌گردد. اکنون، امکان اصلاح این تصویر با استفاده از دیجیتایزر نیز وجود دارد (شکل ۵ – ۱).

شکل ۵ – ۱ نمونه‌ای از دیجیتایزر.

اسکنر، دستگاهی است که تصاویر، نقاشی‌ها یا عکس‌ها را به دیجیتال تبدیل کرده، وارد رایانه می‌کند. در صورتی که بخواهید عکس‌های کاغذی خودتان را از طریق پست الکترونیک برای یکی از دوستان بفرستید، باید آن را از طریق اسکنر به رایانه‌تان انتقال داده، سپس، فایل تصویر را ضمیمه پست الکترونیک نمایید(شکل ۶– ۱).

درایوها، محل‌هایی هستند که دیسک در آن‌ها قرار گرفته تا اطلاعات خاصی را بر روی دیسک نوشته یا از آن بخوانند. در رایانه‌های امروزی درایوهای مختلفی از قبیل درایو CD، درایو دیسک فلاپی، درایو DVD و درایو دیسک سخت وجود دارند. امروزه، از برخی از رایانه‌ها درایوهای فلاپی حذف شده است.

دوربین اینترنتی[3]، بر روی صفحه نمایش رایانه قرار گرفته، تصاویر جلوی رایانه را به آن انتقال می‌دهد (شکل ۷ – ۱).

---

[1].Sensors [2]. Mark Readers [3].Web Cam

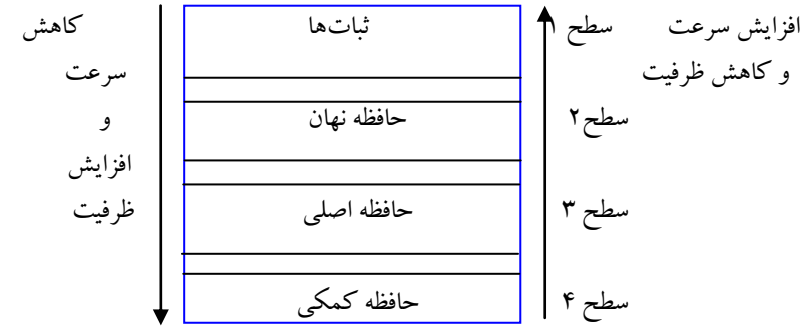
دوربین‌های دیجیتال، دوربین‌هایی هستند که دارای حافظه بوده، تصاویر زیادی را در خود ذخیره می‌نمایند. چون، این دوربین‌ها تصاویر را به صورت دیجیتالی (0 یا 1) ذخیره می‌کنند، **دوربین‌های دیجیتالی** نام دارند. پس از ذخیره تصاویر در دوربین‌های دیجیتالی می‌توان تصاویر را از طریق یک کابل به رایانه انتقال داده و در برنامه‌های گرافیکی نظیر فتوشاپ و فری هند آن‌ها را مشاهده و ویرایش نمود.

۵ – ۱. واحد حافظه

واحد حافظه، برای نگهداری داده‌ها، اطلاعات و برنامه‌ها به کار می‌رود. در شکل ۸ – ۱ سطوح مختلف حافظه را می‌بینید. همان‌طور که در این شکل مشاهده می‌گردد، هر چه به سطوح بالاتر می‌رویم، سرعت حافظه‌ها بیشتر و ظرفیت آن‌ها کاهش می‌یابد (یعنی، با حرکت به سطوح پایین‌تر ظرفیت آن‌ها بیشتر خواهد شد و سرعت آن‌ها کاهش می‌یابد).

شکل ۶– ۱ اسکنرها.

شکل ۷ – ۱ دوربین‌های اینترنتی.



شکل ۸ – ۱ سطوح مختلف حافظه.

ثبات‌ها[1]، حافظه‌هایی هستند که در پردازنده (CPU) قرار دارند (ثبات‌ها ۸، ۱۶، ۳۲ و ۶۴ بیتی هستند). همان‌طور که دیدید، حداکثر ظرفیت آن‌ها ۶۴ بیت است. چون، در CPU قرار دارند، سرعت دسترسی به آن‌ها خیلی بالا است.

[1] Registers

حافظه نهان[1]، نوعی حافظه با دست‌یابی سریع است. این حافظه، آخرین دستورات و اطلاعات پردازش شده را در خود نگهداری می‌نماید تا در صورت نیاز دوباره CPU به آن‌ها، با سرعت (بدون پردازش مجدد) مورد استفاده قرار گیرند. یعنی، دستوراتی که CPU اجرا می‌کند، ابتدا در حافظه نهان آن‌ها را جست‌وجو می‌کند، اگر در این حافظه نباشد، آن‌ها را از حافظه RAM خوانده، اجرا می‌کند. این نوع حافظه‌ها در مقایسه با حافظه اصلی بسیار سریع‌تر، ولی گران‌تر هستند. اکثر رایانه‌های امروزی دارای حافظه نهان هستند.

۱ – ۵ – ۱. حافظه اصلی

دو نوع حافظه اصلی وجود دارد که عبارت‌اند از:

۱. حافظه RAM[2] **(حافظه ناپایدار)**، حافظه با دست‌یابی تصادفی یا خواندنی- نوشتنی است. این حافظه اطلاعات را به صورت موقت نگهداری می‌کند. زیرا، با قطع جریان برق اطلاعات آن حذف خواهد شد. به همین دلیل، این نوع حافظه را **ناپایدار** یا **فرار** می‌نامند. تمام برنامه‌هایی که رایانه اجرا می‌کند باید به این حافظه انتقال یابند. سرعت دسترسی حافظه‌های RAM نانو ثانیه است. در شکل ۹ – ۱ نمونه‌هایی از حافظه RAM را می‌بینید.

۲. حافظه ROM[3] **(پایدار)**، یک حافظه فقط خواندنی است. یعنی، اطلاعات این حافظه قابل تغییر نیست. حتی با قطع جریان برق یا با راه اندازی دوباره رایانه اطلاعات آن از بین نمی‌رود. به همین دلیل، این **حافظه پایدار** نام دارد. اطلاعات حافظه ROM توسط شرکت سازنده در آن نوشته می‌شود. این حافظه شامل دستورالعمل و برنامه‌هایی است که برای راه اندازی مجدد و تست قطعات رایانه به کار می‌روند. مهم‌ترین حافظه ROM، بایوس[4] است که اطلاعات پایداری از قبیل برنامه‌های مربوط به ورودی – خروجی پایه سیستم، بخش‌های اولیه سیستم عامل و مدیریت داخلی رایانه در آن قرار دارند.

۲–۵ – ۱. حافظه کمکی

برای این که دلیل نیاز به حافظه جانبی بیان گردد، کافی است به این سوال پاسخ داده شود که آیا در منزل فقط به اندازه‌ای مواد غذایی نگهداری می‌گردد که الساعه مصرف شود؟ اگر این طور باشد، در این صورت، برای تهیه هر وعده غذا، باید تمام مواد غذایی مورد نیاز را همان موقع خریداری نمود. بدین ترتیب تمام وقتمان صرف تهیه مواد غذایی مورد نیاز همان روز خواهد شد. ولی، معمولاً این کار انجام نمی‌شود، بلکه موادی خریداری می‌گردد که در آینده مورد استفاده قرار می‌گیرد. این مواد را در یخچال نگهداری، در مواقع نیاز از آن‌ها استفاده می‌شود. در استفاده از رایانه نیز چنین وضعیتی وجود دارد. یعنی، اطلاعات مورد نیاز رایانه باید در آن نگهداری شود، هر چند این اطلاعات همان لحظه مورد استفاده قرار نگیرند.

از طرف دیگر، حافظه‌های RAM ناپایدار هستند. یعنی، با قطع جریان برق اطلاعاتشان را از دست می‌دهند. پس، این حافظه‌ها برای نگهداری اطلاعات دائمی کاربردی ندارند (اطلاعات را به صورت موقت نگهداری

---

[1].Cache Memory [2].Random Access Memory [3].Read Only Memory [4].Basic Input Output System (BIOS)

می‌کنند). در ضمن ظرفیت حافظه RAM محدود است و برای انتقال اطلاعات از یک نقطه به نقطه دیگر کارایی ندارد. زیرا، در داخل رایانه قرار دارند و قابل حمل نیستند.

در ضمن، حافظه‌های ROM فقط خواندنی هستند. پس، توسط کاربر قابل تغییر نیستند. اگر هم توسط کاربر قابل تغییر باشند، هر کاربری نمی‌تواند آن‌ها را تغییر دهد. کاربر برای تغییر و برنامه نویسی حافظه‌های ROM از قبیل PROM، EPROM و EEPROM نه‌تنها باید تخصص ویژه‌ای داشته باشد، بلکه باید از تجهیزات خاصی استفاده کند. پس، به حافظه‌های دیگری نیاز است تا:

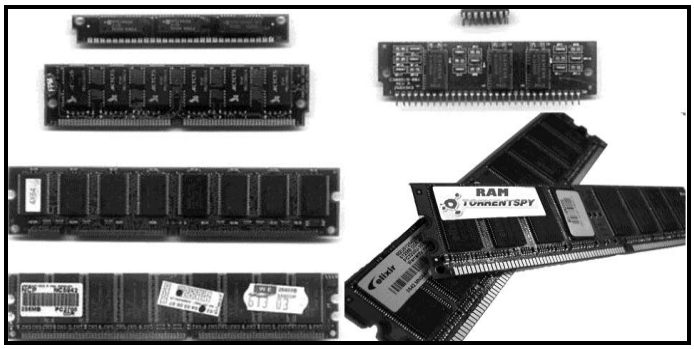
۱. اطلاعات را به صورت دائمی نگهداری کنند.

۲. توسط کاربر قابل تغییر باشند و برای تغییر، به تخصص و دستگاه‌های خاصی نیاز نداشته باشند.

۳. قابل حمل و نقل باشند.

این حافظه، **حافظه ثانویه (جانبی)** نام دارد. حافظه‌های جانبی انواع مختلف دارند که برخی از آن‌ها در زیر آمده‌اند:

۱. دیسک‌های نرم مغناطیسی	۲. دیسک سخت	۳. فلش
۴. دیسک نوری	۵. DVD	۶. نوارهای مغناطیسی
۷. دیسک‌های فشرده	۸. دیسک‌های JAZ	



شکل ۸ – ۱ نمونه‌های از حافظه RAM.

دیسک‌های نرم: دارای یک صفحه مغناطیسی بوده و اطلاعات در دو طرف این دیسک‌ها نوشته می‌شود. این دیسک‌ها، از نوع خواندنی- نوشتنی هستند. ارزان بودن، حجم فیزیکی کم این دیسک‌ها و جدا شدن از گرداننده خودش، موجب شده بود که این دیسک‌ها به عنوان ابزار مناسبی برای حمل اطلاعات در آیند. برخی از رایانه‌های امروزی یک دیسک نرم ۳٫۵ اینچ دارند. چون ظرفیت این دیسک‌ها پایین است، روز به روز استفاده از این دیسک‌ها کم رنگ‌تر خواهد شد.

دیسک‌های سخت مانند دیسک‌های نرم، صفحات مغناطیسی برای ذخیره اطلاعات دارند. دیسک سخت دارای چند صفحه است (شکل ۹ – ۱). صفحات دیسک سخت به صورت ثابت در داخل گرداننده قرار دارند

و مانند فلاپی قابل حمل نیستند. به همین دلیل، نام دیگر این دیسک‌ها، دیسک ثابت است. البته دیسک‌های سخت خارجی وجود دارند که از طریق یک کابل به رایانه متصل شده و قابل انتقال از یک رایانه به رایانه دیگر هستند. برخی از ویژگی‌های دیسک سخت در زیر آمده است:

۱.دارای ظرفیت بالا هستند.

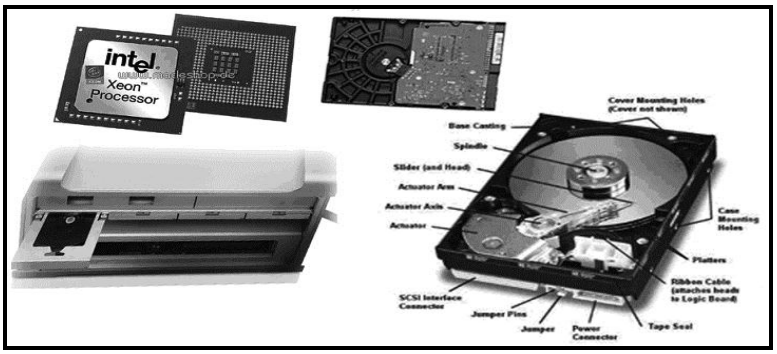
۲.نسبت به دیسک نرم، با توجه به ظرفیت آن‌ها قیمت پایینی دارند.

۳.امکان حمل و نقل دیسک‌های سخت ثابت وجود ندارد.

۴.سرعت زیادی دارند.

۵.قابلیت اطمینان آن‌ها نسبت به دیسک‌های نرم بیشتر است.

امروزه دیسک‌های سخت 320GB، 512GB، 1TB، 2TB و بیشتر از آن وجود دارند.



شکل ۹ – ۱ نمایی از دیسک سخت.

حافظه فلش، حافظه‌های قابل حمل هستند که به دلیل سرعت بالای آن‌ها در ثبت اطلاعات و همچنین استفاده فوق‌العاده آسان بسیار پر فروش و پر طرف دار هستند. از این رو، در دوربین‌های دیجیتالی، تلفن همراه و سایر دستگاه‌ها شاهد استفاده روز افزون از آن‌ها هستیم.

شیوه ذخیره اطلاعات در این نوع از حافظه بسیار شبیه به ذخیره اطلاعات در RAM است. در حقیقت حافظه‌های فلش در نحوه فعالیت مشابه یک منبع ذخیره اطلاعات ثابت عمل می‌کند . یعنی، در آن‌ها هیچ قطعه متحرکی به کار نرفته و تمام کارها توسط مدارات الکترونیکی انجام می‌شود . در مقابل، درون دیسک‌های سخت چندین قسمت متحرک وجود دارد که این وضع خود آسیب پذیر بودن این گونه حافظه را نسبت به حافظه‌های فلش نشان می‌دهد. استفاده از حافظه فلش نسبت به دیسک سخت دارای مزایای زیر است:

۱.حافظه فلش نویز پذیر نیست(جریان مغناطیسی روی داده‌ها تأثیر ندارد).

۲.سرعت دست‌یابی به حافظه‌های فلش بالا است.

۳.حافظه‌های فلش دارای اندازه کوچک هستند.

۴.حافظه فلش دارای عناصر قابل حرکت (نظیر دیسک سخت) نمی‌باشند.

۵.قیمت حافظه‌های فلش نسبت به دیسک سخت کم‌تر است.

لوح‌های فشرده (CD-Rom)، دیسک‌هایی با ظرفیت بالا هستند که برای ذخیره و بازیابی اطلاعات با حجم زیاد به کار می‌روند. لوح‌های فشرده برخلاف دیسک‌های نرم و سخت که اطلاعات را به صورت مغناطیسی می‌نویسند و می‌خوانند، توسط یک پرتو لیزر نوشتن و خواندن اطلاعات را انجام می‌دهند. این پرتوها اثراتی بر روی دیسک ایجاد می کنند که بعداً توسط شعاع لیزر دیگری قابل خواندن هستند. این نوع حافظه‌ها سه دسته هستند:

۱. **CD – ROM**، دیسک‌های فشرده فقط خواندنی هستند. این دیسک‌ها ظرفیت 650MB دارند.

۲. **WOPM**[1]، فقط یک بار قابل نوشتن هستند و چند بار می‌توان از آن‌ها خواند. یعنی، پس از نوشتن، اطلاعات آن قابل تغییر نیست.

۳. **Mo**، اطلاعات را چند بار می‌توان در آن نوشت و چندین بار از آن خواند. یعنی، پس از نوشتن، اطلاعات آن‌ها قابل تغییر است.

دیسک ویدیوی دیجیتالی (DVD)[2] همانند CD است. اما ظرفیت آن‌ها تا ۴/۷۵ گیگابایت می‌باشد. DVDها توسط یک گرداننده (Drive) خوانده یا نوشته می‌شوند. برخی از گرداننده‌ها فقط قدرت خواندن و بعضی دیگر از آن‌ها هم قدرت نوشتن و هم قدرت خواندن دارند. یک لوح فشرده دیجیتالی همانند دیسک نرم قابل پاک کردن اطلاعات قبلی و استفاده مجدد نیست. اما، ظرفیت و قابلیت اطمینان آن نسبت به دیسک‌های نرم بسیار بالاتر است و قیمت مناسبی دارند.

دیسک بلو - ری(Blue-ray Disc یا BD) نام یک نوع دیسک نوری است. در حال حاضر بلو- ری بیشتر برای ذخیره ویدیو با وضوح بسیار بالا به کار می‌رود. دیسک‌های بلو- ری از نظر ظاهری شبیه دیسک‌های DVD و CD هستند و گنجایش‌شان ۲۵ گیگا بایت (یک لایه) یا ۵۰ گیگا بایت (دو لایه) است. DVD وCD با اشعه قرمز کار می‌کنند. اما بلو- ری همان طور که از نامش پیداست، با اشعه آبی کار می‌کند. نام بلو- ری از لیزر آبی بنفشی که در آن برای ذخیره و بازیابی استفاده می‌شود، گرفته شده است. چون این نوع لیزر دارای طول موج کوتاه‌تری (nm۴۰۵) است، در نتیجه داده و اطلاعات بیشتری نسبت به فرمت دیسک‌های DVD که دارای لیزر قرمز (nm۶۵۰) هستند، را می‌توان روی دیسک‌های بلو- ری ذخیره نمود. یک بلو- ری دو لایه توانایی ذخیره ۵۰ گیگا بایت، بیشتر از ۵ برابر ظرفیت یک دیسک‌های DVD دو لایه را دارد(شکل ۱۰-۱).

شکل ۱۰- ۱ نمونه دیسک بلو- ری.

۳-۵ - ۱. واحدهای اطلاعاتی و حافظه

واحدهای مختلفی برای اندازه‌گیری ظرفیت حافظه وجود دارند که کوچک‌ترین آن‌ها **بیت**[3] است. هر بیت می‌تواند یک مقدار ۰ یا ۱ را در خود نگهداری کند. هر ۸ بیت یک **بایت**[4] نام دارد. حداکثر ۲ به توان ۸ (۲۵۶)

---

[1]. Write Once Read Many [2].Digital Versatile Disk [3].Bit [4].Byte

ترکیب مختلف(نماد) را می‌توان در یک بایت قرار داد. **کلمه**[1]، به مجموعه دو یا چهار بایتی گفته می‌شود که از یک ماشین به ماشین دیگر ممکن است تفاوت داشته باشد. **کلمه مضاعف**[2]، به هر دو کلمه گفته می‌شود. کیلوبایت، به مجموعه‌ی ۱۰۲۴ ($۲^{۱۰}$) بایت گفته می‌شود. انواع واحدهای اندازه‌گیری ظرفیت حافظه را در جدول ۱ -۱ می‌بینید.

حافظه مجازی، بعضی از برنامه‌ها و فایل‌ها ممکن است آن قدر بزرگ باشند که در حافظه اصلی جا نشوند و نیاز به فضای بیشتری از حافظه اصلی داشته باشند. در این صورت، در سیستم‌های عامل پیشرفته نظیر ویندوز، لینوکس و یونیکس، بخشی از برنامه در یک حافظه کمکی مانند دیسک سخت قرار می‌گیرد. این نوع حافظه‌ها، **حافظه مجازی**[3] نام دارند. سیستم‌های عاملی که از حافظه مجازی استفاده می‌کنند، فقط بخشی از برنامه که در اجرا مورد نیاز هستند، در حافظه اصلی، و بقیه در حافظه مجازی قرار می‌گیرند. در این صورت، به علت پایین بودن سرعت حافظه کمکی (حافظه مجازی در نظر گرفته شده) نسبت به حافظه RAM، سرعت اجرای برنامه نیز کاهش می‌یابد.

جدول ۱ - ۱ واحدهای اندازه‌گیری‌های مختلف برای ظرفیت حافظه.

نام	مخفف	
Kilo (کیلو)	KB	$۲^{۱۰}$ = ۱۰۲۴ بایت
Mega (مگا)	MB	۱۰۲۴KB یا ۱۰۴۸۵۷۶ = $۲^{۲۰}$ بایت
Giga (گیگا)	GB	۱۰۲۴MB یا ۱۰/۷۳/۷۴۱/۸۲۴ = $۲^{۳۰}$ بایت
Tera (ترا)	TB	۱/۰۹۹/۵۱۱/۶۲۷/۷۷۶ = $۲^{۴۰}$ بایت
Peta (پتا)	PB	۱/۱۲۵/۸۹۹/۹۰۶/۸۴۲/۶۲۴ = $۲^{۵۰}$ بایت
Exa (اکسزا)	EB	۱/۱۵۲/۹۲۱/۵۰۴/۶۰۶/۸۴۶/۹۷۶ = $۲^{۶۰}$ بایت
Zetta (زتا)	ZB	۱/۱۸۰/۵۱۹/۶۲۰/۷۱۷/۴۱۱/۳۰۳/۴۲۴ = $۲^{۷۰}$ بایت
Yotta (یوتا)	YB	۱/۲۰۸/۹۲۵/۸۱۹/۶۱۴/۶۲۹/۱۷۴/۷۰۶/۱۷۶ = $۲^{۸۰}$ بایت

۶ - ۱. واحد خروجی

واحد خروجی اطلاعات موجود در رایانه را در اختیار کاربران قرار می‌دهد. این کار از طریق دستگاه‌های خروجی انجام می‌شود. در این بخش برخی از دستگاه‌های خروجی و وظیفه آن‌ها را می‌بینید.

صفحه نمایش رایج‌ترین دستگاه خروجی است. نام دیگر صفحه نمایش، مانیتور[4] می‌باشد. صفحه نمایش، برای نمایش اطلاعات به کار می‌رود. انواع صفحه نمایش‌ها عبارت‌اند از: صفحه نمایش با لامپ اشعه کاتد یک (CRT)[5]، صفحه نمایش کریستال مایع (LCD)[6] نمایشگرهای LCS[7]، نمایشگرهای Super LCD، نمایشگرهای LED[8]، نمایشگرهای AMOLED[9] و نمایشگرهای سه بعدی و عینک سه بعدی.

---

[1].Word [2].Double Word [3].Virtual Memory [4].Monitor [5].Cathode Ray Tube
[6].Liquid Crystal Display [7].Liquid Crystal Shutter [8].Light Emitting Diode [9].Active Matrix

چاپگر، متداول ترین دستگاه خروجی برای چاپ اطلاعات در رایانه است. چاپگرها دو نوع دارند که عبارت‌اند از:

۱. چاپگرهای ضربه‌ای (Impact Printers). **۲.** چاپگرهای غیر ضربه‌ای (NoImpact Printers).

چاپگرهای ضربه‌ای: در این نوع چاپگرها، عمل چاپ با ضربه به ریبون[1] توسط هد چاپگر بر روی کاغذ انجام می‌گردد. این چاپگرها دو نوع هستند که عبارت‌اند از :**چاپگرهای با کاراکترهای از پیش تعریف شده و چاپگرهای ماتریسی**[2].

چاپگرهای غیر ضربه‌ای: این چاپگرها، از طریق حرارت، جوهر افشانی (پاشیدن) و تونر[3] کاراکترها را به کاغذ منتقل می‌کنند. این چاپگرها سریع تر بوده و در هنگام چاپ صدایی ندارند. انواع مختلف چاپگرهای غیر ضربه‌ای عبارت‌اند از: **چاپگرهای جوهر افشان، چاپگرهای لیزری تک رنگ** و **چاپگرهای لیزری رنگی.**

رسام[4]، برای چاپ تصاویر و نقشه‌های طراحی شده توسط رایانه به کار می‌رود (شکل ۱۱ – ۱). سرعت رسام پایین، ولی دقت آن بسیار خوب است. رسام می‌تواند لیزری یا جوهر افشان باشد. رسام‌ها سه نوع دارند که عبارتند از: **رسام مسطح، رسام غلتکی و رسام ترکیبی.**

۱۱- ۱ نمونه دیسک رسام.

بلندگو[5]، به کارتی به نام کارت صدا در رایانه متصل شده، صدای برآمده از رایانه را پخش می‌کند. امروزه صفحه نمایش‌هایی وجود دارند که بلندگو در جلوی آن تعبیه شده است. از طریق این صفحه نمایش‌ها نیز می‌توان صدای حاصله را پخش نمود.

عینک واقعیت مجازی، برای ایجاد تصاویر سه‌بعدی به کار می‌رود. کاربر با قرار دادن این عینک در چشم خودش، می‌تواند تصاویر را به شکل سه‌بعدی مشاهده کند.

ویدیو پروژکتور رومیزی[6]، دستگاهی است که به رایانه (خروجی رایانه) متصل می‌گردد و آنچه را که در صفحه نمایش دیده می‌شود، را به صورت بزرگ‌تر روی صفحه سفید (روی دیوار) نمایش می‌دهد. امروزه در اکثر کلاس‌ها و سمینارهای آموزشی از ویدیو پروژکتور به جای تخته سیاه استفاده می‌گردد.

موتورهای کنترل شونده[7]، شبیه آرمیچرهایی هستند که حرکت آن‌ها به سمت چپ و راست، در درجات و اندازه‌های مختلف، توسط رایانه قابل کنترل است. به عنوان مثال، حرکت کمتر از یک درجه به سمت خاص، حرکت روبات‌ها و بازوهای آن‌ها توسط موتورهای کنترل شونده انجام می‌گیرد.

رسانه‌های انتقال اطلاعات، با استفاده از دیسک، فلش، CD، DVD و شبکه می‌توان اطلاعاتی را از یک رایانه به یک رایانه دیگر انتقال داد. رسانه‌های انتقال اطلاعات می‌توانند هم به عنوان ورودی و هم به عنوان خروجی اطلاعات در رایانه به کار روند. امروزه مهم‌ترین رسانه‌های انتقال اطلاعات فلش و پست الکترونیکی هستند.

---

[1].Ribbon [2].Matrix Printers [3].Toner [4].Platter [5]. Speaker [6].Desktop Projector
[7].Stepper Motors [8].Modem

مودم[۸]، وسیله‌ای ورودی و خروجی است که وظیفه‌اش مبادله اطلاعات بین دو رایانه از طریق خط تلفن می‌باشد. مودم، برای کاربردهای مانند اتصال به اینترنت استفاده می‌شود.

۷ – ۱. انتقال اطلاعات و پیدایش شبکه

اتصال رایانه‌ها به یک دیگر را شبکه[۱] گویند. رایانه‌ها در یک شبکه یا **سرویس‌دهنده**[۲] هستند یا **سرویس گیرنده**[۳]. **سرویس‌دهنده**، درخواست‌ها را از **سرویس گیرنده‌ها** دریافت کرده، پردازش‌های لازم را انجام داده به سرویس گیرنده برمی‌گرداند. نام دیگر سرویس گیرنده **گره**[۴] یا **ایستگاه کاری**[۵] است. مهم‌ترین اهداف شبکه، استفاده هم‌زمان و اشتراکی رایانه‌ها از منابع اطلاعاتی مانند بانک اطلاعاتی، منابع سخت افزاری نظیر چاپگرها، دیسک‌های سخت و غیره، تبادل اطلاعات بین رایانه‌ها و کاهش هزینه است.

شبکه‌ها را بر حسب فواصل فیزیکی رایانه‌ها از یک دیگر به دو دسته زیر تقسیم می‌کنند:

شبکه محلی(LAN) ، در محدوده کوچکی مانند یک ساختمان مورد استفاده قرار می‌گیرند.

شبکه‌های گسترده (WAN)[۶]، در محدوده وسیعی نظیر شهرها و کشورها مورد استفاده قرار می‌گیرند. اینترنت نمونه‌ای از شبکه گسترده است.

با وجود پیشرفت‌های زیادی که در زبان‌های برنامه‌نویسی و متدهای برنامه‌نویسی صورت گرفته است، بر خلاف تصور برخی از برنامه‌نویسان مبتدی یا غیرحرفه‌ای که معتقدند عمر الگوریتم و فلوچارت به پایان رسیده، باید متذکر شد که هنوز از این ابزار برای طراحی برنامه استفاده می‌شود و برنامه‌نویسان حرفه‌ای در پروژه‌های بزرگ، طراحی برنامه‌ها را به وسیله الگوریتم یا فلوچارت (و یا ابزارهایی مشابه آن) انجام می‌دهند. بنابراین، لازم است با این مفاهیم به طور کامل آشنا شویم و معتقدیم بدون درک صحیح این مفاهیم، برنامه‌نویسی با مشکلات و مسائل زیادی همراه است.

۸ – ۱. مراحل برنامه‌نویسی

برنامه‌نویسی دارای مراحل زیر است:

۱. **شناخت مسئله**، که شامل مراحل زیر می‌باشد:

- شناخت فرضیات و داده‌های ورودی مسئله (Data)
- تشخیص خواسته‌ها یا مجهولات مسئله
- تعیین ارتباط بین فرضیات و مجهولات

۲. **ارائه طرح یا نقشه حل مسئله**

روش‌های مختلفی برای حل مسئله وجود دارد. برخی از این روش‌ها عبارت‌اند از:

- طراحی الگوریتم
- طراحی فلوچارت

---

[1].LAN (Local Area Network) [2].Data [3].Process [4].Information [5].Feedback [6].Wide Area Network

۳. نوشتن برنامه با یکی از زبان‌های برنامه‌نویسی (نظیر C، پاسکال، C++ یا C#).

۹– ۱. الگوریتم چیست؟

وقتی می‌خواهید کاری را انجام دهید، به ویژه اگر آن کار پیچیده باشد، ابتدا، باید روش و مراحل انجام آن کار را مشخص نمایید. به عنوان مثال، فرض کنید مسئول برگزاری جشنی هستید، در این صورت باید خودتان را برای انجام کارهای لازم جهت برگزاری این جشن آماده کنید. در آن صورت چه می‌کنید؟ مطمئناً لیستی از کارهایی که باید انجام دهید، تعیین کرده، ترتیب انجام آن‌ها را مشخص می‌کنید. یعنی، روش و مراحل انجام کار را به دقت تنظیم خواهید کرد. چنانچه این کار را به نحو کامل و بدون **نقص** انجام دهید که تحت هر شرایطی با هر نوع امکانات و خصوصیات بتوان از این رویه استفاده نمود، اکنون توانسته‌اید الگوریتم برگزاری مراسم را طراحی کنید. حتی می‌توانید این الگوریتم را به دیگران بدهید تا آن‌ها نیز بتوانند از این الگوریتم برای برگزاری جشن‌شان استفاده کنند.

الگوریتم، عبارت است از مجموعه‌ای از دستورالعمل‌ها که مراحل انجام کاری را **به زبان دقیق** با **جزئیات کافی** بیان نماید، ترتیب **اجرای دستورات** و **شرط خاتمه** آن مشخص باشد.

برای این که با مفهوم الگوریتم آشنا شویم، دستورالعمل استفاده از تلفن همگانی را در نظر بگیرید. این دستورالعمل در زیر آمده است:

۱. گوشی را بردارید.
۲. یک سکه در داخل تلفن قرار دهید.
۳. منتظر شنیدن بوق آزاد باشید.
٤. شماره را گرفته، صحبت کنید.
٥. در پایان، گوشی را در جایش قرار دهید.

مراحل بیان شده، دستورالعمل استفاده از تلفن همگانی است (الگوریتم نیست). زیرا:

۱. مرحله ۲ به زبان دقیق بیان نشده است. یعنی، استفاده کننده تلفن نمی‌داند چه سکه‌ای را در داخل تلفن قرار دهد. برای این که این مرحله به یک دستورالعمل الگوریتم تبدیل شود، این دستورالعمل باید به صورت زیر تغییر یابد:

- یک سکه ۲۵۰ یا ۱۰۰ ریالی سالم در داخل تلفن قرار دهید.

۲. شرط خاتمه الگوریتم مشخص نشده است. زیرا، اگر تلفن خراب باشد، در مرحله ۳ هیچ‌گاه بوق آزاد شنیده نمی‌شود. بنابراین، استفاده‌کننده تلفن تا کی منتظر شنیدن بوق آزاد باشد. این مرحله را به صورت زیر تغییر دهید تا به یک مرحله الگوریتم تبدیل شود:

۳۰ ثانیه منتظر شنیدن بوق آزاد باشید. چنانچه بوق آزاد را نشنیده‌اید، گوشی را در جایش قرار دهید. مراحل ۱ تا ۳ را اجرا کنید. چنانچه ۳ بار این مراحل را انجام داده‌اید و بوق آزاد را نشنیده‌اید، گوشی را در جایش قرار دهید.

بنابراین، الگوریتم استفاده از تلفن همگانی به صورت زیر است:

۱. شروع
۲. گوشی را بردارید.
۳. یک سکه ۱۰۰ یا ۲۵۰ ریالی سالم در داخل تلفن قرار دهید.
٤. ۳۰ ثانیه منتظر شنیدن بوق آزاد باشید، چنانچه بوق آزاد را نشنیده‌اید، گوشی را در جایش قرار دهید. مراحل ۱ تا ۴ را اجرا کنید. اگر ۳ مرتبه این مراحل را انجام داده‌اید و بوق آزاد را نشنیده‌اید، گوشی را جایش قرار دهید و دنبال تلفن دیگر بگردید.
۵. شماره‌گیری نموده و صحبت کنید
٦. در پایان گوشی را جایش قرار دهید.
۷. پایان

مثال ۱ – ۱. الگوریتمی که مراحل آماده‌سازی نیمرو را بیان می‌کند.

۱. شروع
۲. ماهی‌تابه را بردارید.
۳. در ماهی‌تابه روغن بریزید.
٤. اجاق را با حرارت ملایم روشن نمایید.
۵. منتظر داغ شدن روغن بمانید.
٦. پس از داغ شدن روغن، تخم مرغ را در ماهی‌تابه بشکنید.
۷. به تخم مرغ نمک و فلفل اضافه نمایید.
۸. تا زمانی که نیمرو به اندازه کافی بپزید، صبر کنید.
۹. اجاق را خاموش نمایید.
۱۰. نیمرو را در بشقاب قرار دهید.
۱۱. پایان.

همان‌طور که در این الگوریتم مشاهده می‌کنید، هر الگوریتم یک **نقطه شروع** و یک **نقطه پایان** دارد.

۱۰ – ۱. تفکر الگوریتمیک

آیا مفهوم الگوریتم فقط در دنیای برنامه‌نویسی تعریف می‌شود؟ آیا تنها برنامه‌نویس برای طراحی و حل برنامه‌اش باید از الگوریتم استفاده کند؟ یا این که برای انجام تمام کارها، می‌توان از ایده الگوریتمی استفاده نمود؟

در جواب این سوالات باید بگوییم که الگوریتم یک مفهوم عام و فراگیر است. یعنی، ایده الگوریتم، علاوه بر آن که در برنامه‌نویسی به کار می‌رود، می‌تواند در زندگی روزمره نیز استفاده شود.

یک برنامه‌نویس خوب، علاوه بر آن که به الگوریتم برنامه‌هایی که می‌نویسد فکر می‌کند، با الگوریتم زندگی می‌کند. او تمام زندگی خودش را با الگوریتم انجام می‌دهد. با الگوریتم فکر می‌کند. یعنی، تفکر او حتی در امور ساده زندگی نیز الگوریتمیک است. این فرد احتمالاً برنامه‌نویس موفقی خواهد بود. این تنها کافی نیست که بخواهید الگوریتم یک کار را پیدا کنید. بلکه ماهیتاً باید الگوریتمیک فکر کنید. حتی به تمام مسائل زندگی نیز باید به این شیوه نگاه کنید. مثال برنامه نویس که دارای تفکر الگوریتمیک است، نسبت به فردی که این ویژگی را ندارد، مانند نسبت کسی است که ماهیتاً کارش دارای نظم است و کسی که کارش نظم خاصی ندارد. فرد اول، در پایان کار، کارگاه خود را مرتب و منظم می‌کند. اما، کارگاه فرد دوم نه تنها در هنگام کار نامرتب است، بلکه بعد از انجام کار نیز نامنظم خواهد بود.

> وقتی می‌خواهید کاری را انجام دهید، باید به الگوریتم مناسب برای انجام آن کار فکر کنید. پس از یافتن الگوریتم مناسب، انجام آن کار برای‌تان بسیار ساده خواهد شد. چیزی را فراموش نمی‌کنید و در انجام آن کار، همه ابعاد آن را در نظر خواهید گرفت و مهم‌تر از همه این‌ها، ساختار کار برای شما روشن خواهد بود. برنامه‌نویس باید با الگوریتم زندگی کند و آن را به عنوان روش زندگی خود بشناسد.
>
> ارائه راه حل مناسب، نیاز دقیق به شناخت مسئله دارد.

مثال ۲ – ۱. الگوریتمی که سه عدد را خوانده، میانگین آن‌ها را نمایش می‌دهد.

۱. شروع

۲. a، b و c را بخوان — **داده (۳ عدد) :** a , b , c

۳. $ave \leftarrow \frac{a+b+c}{3}$ — **خواسته** (میانگین) : ave

رابطه : $ave \leftarrow \frac{a+b+c}{3}$

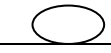
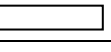
۴. ave را چاپ کن

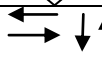
۵. پایان

۱۱ – ۱. تعریف فلوچارت

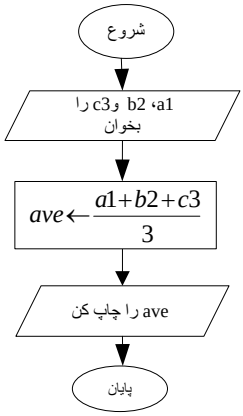
به ارائه‌ی یک طرح منطقی با جزئیات کافی، دقیق و رعایت ترتیب مراحل اجرا، از روش حل یک مسأله فلوچارت (نمودار عملیاتی) آن مسأله می‌گویند. ارائه فلوچارت به کمک علائم قراردادی انجام می‌شود علائم رسم فلوچارت در جدول ۱ – ۱ آمده‌اند.

جدول ۱ - ۱ علائم رسم فلوچارت

شکل	توضیح
	**بیضی:** دستورات شروع و پایان
	متوازی‌الاضلاع: دستورات ورودی و خروجی را تعیین می‌کند.
	**مستطیل:** برای دستورات محاسباتی یا جایگزینی استفاده می‌شود.

	لوزی: برای دستورات شرطی یا تصمیم‌گیری استفاده می‌شود.
	**فلش‌های جهت‌دار:** جهت تعیین مسیر عملیات استفاده می‌شوند.
	دایره کوچک: رابط

به عنوان مثال، فلوچارت میانگین سه عدد در زیر آمده است:



مثال ۳ –۱. الگوریتم و فلوچارتی که ابتدا اضلاع مثلثی را از ورودی خوانده، مساحت آن را محاسبه و نمایش می‌دهد.

داده : a, b, c (**اضلاع مثلث**)

خواسته : s (**مساحت**)

رابطه : (**قاعده هرون**)

$$s = \sqrt{p(p-a)(p-b)(p-c)} \qquad p = \frac{a+b+c}{2}$$

۱. شروع

۲. اضلاع مثلث را بخوان (a و b , c را بخوان)

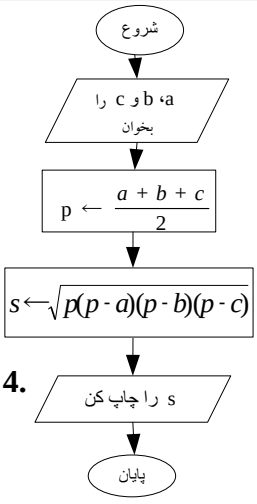
۳. $p \leftarrow \frac{a+b+c}{2}$

p = (x + y) * 2

۵. $s \leftarrow \sqrt{p(p-a)(p-b)(p-c)}$

۶. s را چاپ کن

۷. پایان

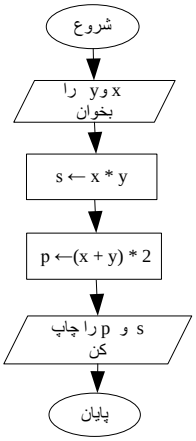


مثال ۴ – ۱. الگوریتم و فلوچارتی که طول و عرض مستطیل را خوانده، محیط و مساحت آن را نمایش می‌دهد.

داده : x و y (**طول و عرض مستطیل**)

خواسته: p و s (**محیط و مساحت مستطیل**)

رابطه: s = x * y و p = (x + y) * 2



۱. شروع
۲. x و y را بخوان
۳. x ضرب در y را در s قرار بده (s ← y * x)
۴. x به علاوه y ضرب در ۲ را در p قرار بده (p ←2 *(x + y)).
۵. s و p را چاپ کن.
۶. پایان

۱۲ – ۱. ساختار تصمیم

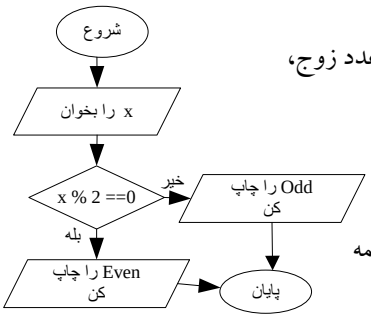
گاهی نیاز است درست[1] یا نادرست[2] بودن شرطی بررسی شود. اگر شرط درست بود، دستور یا مجموعه‌ای از دستورات اجرا شوند، در غیر این صورت، دستور دیگر یا مجموعه‌ای از دستورات دیگر اجرا گردند. برای این منظور، از ساختار تصمیم استفاده می‌شود. در الگوریتم، ساختار تصمیم با **اگر** بیان می‌گردد و در فلوچارت این ساختار با لوزی نمایش داده می‌شود.

مثال ۵ – ۱. الگوریتم و فلوچارتی که عددی صحیح را خوانده، تشخیص می‌دهد زوج است یا خیر؟

داده: x (عدد خوانده شده)

خواسته: چاپ زوج (Even) یا فرد (Odd)

رابطه: اگر باقی مانده تقسیم صحیح x به ۲ برابر صفر باشد، عدد زوج، وگرنه فرد است.



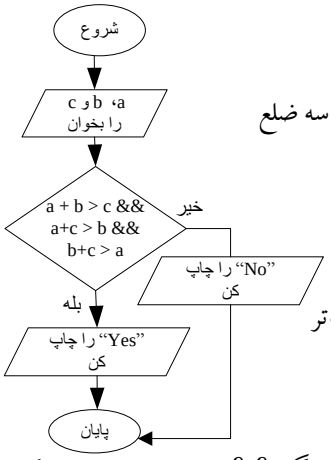
۱. شروع
۲. x را بخوان
۳. اگر باقی مانده تقسیم صحیح x به ۲ برابر با صفر است، کلمه "Even" را چاپ کن. وگرنه، کلمه "Odd" را چاپ کن.
۴. پایان

همان‌طور که در فلوچارت می‌بینید، باقی مانده تقسیم صحیح با عملگر بر نمایش داده می‌شود.

مثال ۶ – ۱. الگوریتم و فلوچارتی که سه عدد را خوانده، تعیین می‌کند که آیا سه عدد تشکیل مثلث می‌دهند یا خیر؟

داده: a، b و c (سه عدد)

[1] .True [2]. False



خواسته: تعیین این که سه عدد تشکیل مثلث می‌دهند یا خیر.

رابطه: هرگاه مجموع هر دو ضلع بزرگ‌تر از ضلع سوم باشد، آن سه ضلع تشکیل مثلث را می‌دهند.

۱. شروع

۲. a ، b و c را بخوان

۳. اگر a + b بزرگ‌تر از c و c + b بزرگ‌تر از a و a + c بزرگ‌تر از b باشد، آنگاه "Yes" را چاپ کن. وگرنه، "No" را چاپ کن.

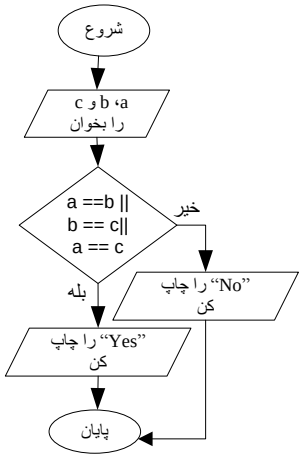
۴. پایان

هدف این الگوریتم آشنایی با عملگر && (و منطقی) است. نتیجه عملگر && زمانی درست است که همه شرط‌ها درست باشند.

مثال ۷ – ۱. الگوریتم و فلوچارتی که سه ضلع مثلثی را خوانده تشخیص می‌دهد آیا این سه ضلع تشکیل مثلث متساوی الساقین را می‌دهند یا خیر؟

داده: a، b و c (سه ضلع مثلث)

خواسته: تعیین این که آیا سه ضلع تشکیل مثلث متساوی‌الساقین می‌دهند یا خیر.



رابطه: a = b یا b = c یا a = c

۱. شروع

۲. a، b و c را بخوان

۳. اگر a برابر b یا b برابر c یا a برابر c آنگاه "Yes" را چاپ کن وگرنه، "No" را چاپ کن.

٤. پایان

هدف این برنامه آشنایی با عملگر || (یا منطقی) است. نتیجه این عملگر زمانی درست است که حداقل یکی از شرط‌ها درست باشند.

۱۳ – ۱. ساختار حلقه

در برخی از برنامه‌ها نیاز است مجموعه‌ای از دستورات چند بار تکرار شوند. برای تکرار این دستورات از حلقه‌های تکرار استفاده می‌شود. زیرا:

۱. از نوشتن دستورات تکراری جلوگیری می‌کند.

۲. از تعریف متغیرهای اضافی جلوگیری خواهد کرد.

مثال ۸ – ۱. الگوریتمی را در نظر بگیرید که بخواهد اعداد ۱ تا ۵ را نمایش دهد. برای این منظور، سه روش وجود دارد:

۱. بدون استفاده از حلقه تکرار، این الگوریتم به صورت زیر است:

۱. شروع
۲. یک را در i قرار بده
۳. i را چاپ کن.
٤. 1 + i را در i قرار بده
۵. i را چاپ کن.
٦. 1 + i را در i قرار بده
۷. i را چاپ کن.
۸. 1 + i را در i قرار بده
۹. i را چاپ کن.
۱۰. 1 + i را در i قرار بده
۱۱. i را چاپ کن.
۱۲. پایان

همان‌طور که می‌بینید، اگر تعداد تکرار زیاد باشد (به عنوان مثال، اگر الگوریتم بخواهد اعداد ۱ تا ۱۰۰ را نمایش دهد)، این فرآیند طولانی و غیر قابل انجام خواهد شد (یعنی، کدهای برنامه زیاد می‌شود).

۲. استفاده از متغیرهای جداگانه، در این روش برای هر یک از مقادیر ۱ تا ۵ متغیر مجزایی در نظر گرفته می‌شود. این الگوریتم در زیر آمده است.

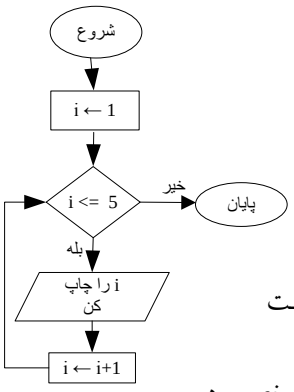
۱. شروع
۲. ۱ را در a، ۲ را در b، ۳ را در c، ۴ را در d و ۵ را در e قرار بده.
۳. a، b، c، d و e را چاپ کن.
٤. پایان.

در این روش اگر تعداد تکرار زیاد باشد، نیاز به متغیرهای زیادی است. بنابراین، حافظه زیادی مصرف خواهد شد.

۳. استفاده از حلقه تکرار، برای ایجاد حلقه تکرار اعمال زیر باید انجام شود:

۱. یک شمارنده در نظر گرفته شود.
۲. مقدار اولیه حلقه در شمارنده قرار گیرد.
۳. شرط ادامه حلقه تست گردد (ابتدای حلقه). اگر شرط ادامه حلقه درست باشد، گام‌های ۴ تا ۶ اجرا گردد، وگرنه حلقه خاتمه یابد.
٤. دستورات بدنه حلقه اجرا شوند.
۵. شمارنده به اندازه گام افزایش، اضافه شود.
٦. به ابتدای حلقه برود (به مرحله ۳ برود).

بنابراین، الگوریتم مثال ۸ – ۱ با استفاده از حلقه تکرار به صورت زیر است:



۱. شروع
۲. یک را در i قرار بده (i شمارنده حلقه است).
۳. تا وقتی که 5 => i تکرار کن
الف: i را چاپ کن
ب: 1 + i را در i قرار بده.
۴. پایان

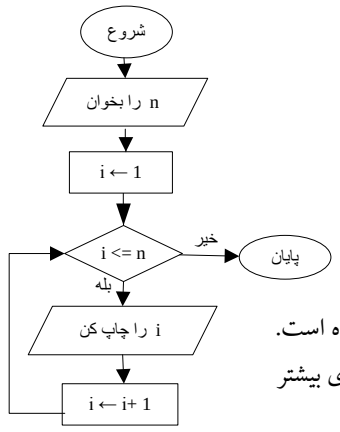
این روش، نه تنها از یک متغیر برای نوشتن برنامه استفاده کرده است (شمارنده i)، بلکه از تکرار کدهای اضافی نیز جلوگیری نموده است.

همان‌طور که مشاهده کردید، در الگوریتم مثال ۸ – ۱ تعداد تکرار از قبل مشخص بود. گاهی اوقات اتفاق می‌افتد که تعداد تکرار متغیر است (مشخص نیست) و بستگی به عددی دارد که کاربر وارد کرده است. در این صورت حلقه، همان حلقه قبلی می‌باشد، با این تفاوت که در شرط حلقه عدد وارد شده قرار می‌گیرد (مثال زیر را ببینید).

مثال ۹ – ۱. الگوریتم و فلوچارتی که عدد مثبتی را خوانده، اعداد ۱ تا عدد خوانده شده را نمایش می‌دهد.

داده: n (عدد خوانده شده) **خواسته:** نمایش ۱ تا n

رابطه: ایجاد حلقه‌ایی با شمارنده i که مقادیر ۱ تا n را می‌پذیرد.



۱. شروع
۲. n را بخوان
۳. ۱ را در i قرار بده
۴. تا وقتی که n => i تکرار کن
الف: i را چاپ کن
ب: 1 + i را در i قرار بده
۵. پایان

این حلقه‌هایی که مشاهده کردید، گام افزایش آن‌ها یک بوده است. گاهی ممکن است گام افزایش حلقه عدد یک نباشد. یعنی، عددی بیشتر یا کمتر از یک باشد (مثال ۱۰– ۱ را ببینید).

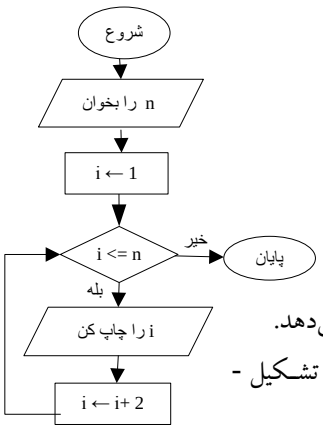
مثال ۱۰ – ۱. الگوریتمی که عددی مثبت را خوانده، اعداد فرد ۱ تا عدد خوانده شده (۱، ۳، ...، n) را نمایش می‌دهد.

داده: n (عدد خوانده شده)

خواسته: نمایش اعداد فرد ۱ تا n

رابطه: تشکیل حلقه تکرار از ۱ تا n به طوری که در هر مرحله ۲ واحد به شمارنده آن اضافه می‌شود.

۱. شروع
۲. n را بخوان
۳. ۱ را در i قرار بده
٤. تا وقتی که i <= n تکرار کن
الف: i را چاپ کن
ب: i + 2 را در i قرار بده
۵. پایان



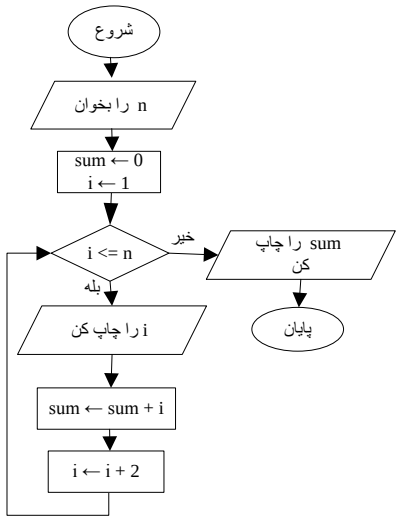
این الگوریتم مانند الگوریتم ۹ – ۱ است. با این تفاوت که در گام پنجم به جای این که i + 1 در i قرار گیرد، i + 2 را در i قرار می‌دهد. حلقه‌هایی که تاکنون دیدید، بدنه حلقه آن‌ها از یک دستور تشکیل - می‌گردیدند.

بدنه حلقه می‌تواند دارای بیش از یک دستور باشد (الگوریتم‌های زیر را ببینید):

مثال ۱۱ – ۱. الگوریتم و فلوچارتی که عددی مثبت را خوانده، اعداد فرد ۱ تا عدد خوانده شده (n) و مجموع آن‌ها را نمایش می‌دهد.

توضیح: در این الگوریتم باید مجموع اعداد فرد نیز نمایش داده شود. برای این منظور، یک ظرف به نام sum در نظر گرفته می‌شود که در ابتدا صفر در آن قرار می‌گیرد. یعنی مجموع اعداد فرد در ابتدا صفر است.

داده : n (عدد خوانده شده)
خواسته: نمایش اعداد فرد ۱ تا n و مجموع آن‌ها
رابطه: ایجاد یک حلقه و استفاده از متغیر sum برای نگهداری مجموع

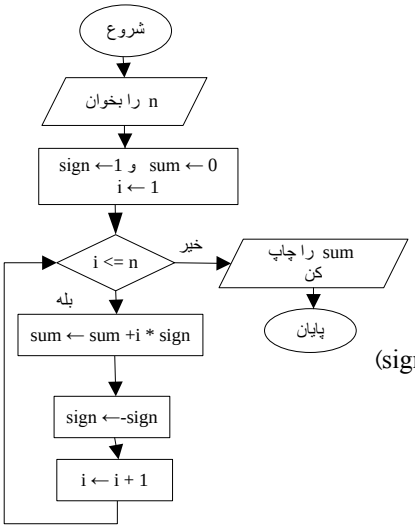


۱. شروع
۲. n را بخوان
۳. صفر را در sum قرار بده
٤. یک را در i قرار بده
۵. تا وقتی که i <= n تکرار کن
الف: i را چاپ کن
ب: i را با sum جمع کرده، در sum قرار بده
ج: i + 2 را در i قرار بده
٦. sum را چاپ کن
۷. پایان

مثال ۱۲ – ۱. الگوریتم و فلوچارتی که n را خوانده، مجموع سری زیر را چاپ می‌کند:

sum = 1 – 2 + 3 - … ± n

توضیح: در این الگوریتم، علامت سری یکی در میان مثبت و منفی می‌شود. بنابراین، یک متغیر به نـام sign در نظر می‌گیریم و در ابتدا مقدار یک را در آن قرار می‌دهیم (چون اولین جملـه علامـت آن مثبـت اسـت). در داخل حلقه sign را در یک منفی ضرب می‌کنیم و در sign قرار می‌دهیم تا یکی در میان ۱+ و ۱– شود.

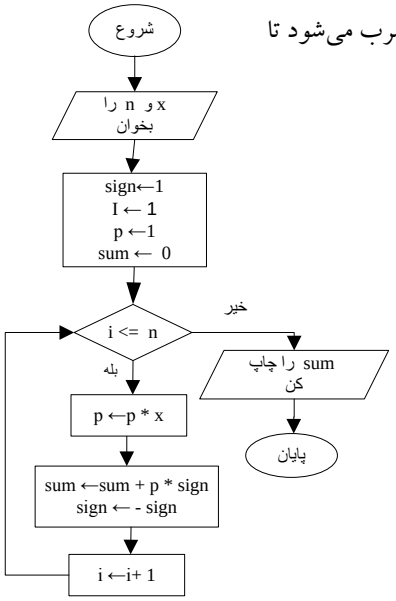


۱. شروع
۲. n را بخوان
۳. یک را در sign قرار بده
٤. صفر را در sum قرار بده
۵. یک را در i قرار بده
٦. تا وقتی که i <= n تکرار کن
الف: sum + i * sign را در sum قرار بده
ب: منفی sign را در sign قرار بده (sign ← sign * -1)
ج: i + 1 را در i قرار بده
۷. sum را چاپ کن
۸. پایان

مثال ۱۳ – ۱. الگوریتم و فلوچارتی که x و n را خوانده، مجموع سری زیر را نمایش می‌دهد:

$$\mathbf{sum = x - x^2 + x^3 - x^4 + \ldots \pm x^n}$$

توضیح: این الگوریتم مانند الگوریتم قبل می‌باشد. با این تفاوت که در داخـل حلقـه بـه جـای ایـن کـه i در sign ضرب گردد، توان‌های از x باید در sign ضرب شود. بنابراین، متغیری برای توان نگهداری x باید در نظر گرفته شود که در ابتدا مقدار آن ۱باشد و در داخل حلقه در x ضرب می‌شود تا توان‌های x ایجاد گردد.



۱. شروع
۲. x و n را بخوان
۳. یک را در p قرار بده (p: متغیر توان x می‌باشد)
٤. sing ← 1
۵. صفر را در sum قرار بده
٦. یک را در i قرار بده
۷. تا وقتی که i <= n تکرار کن
الف: p * x را در p قرار بده
ب: sum + p * sign را در sum قرار بده
ج: منفی sign را در sign قرار بده

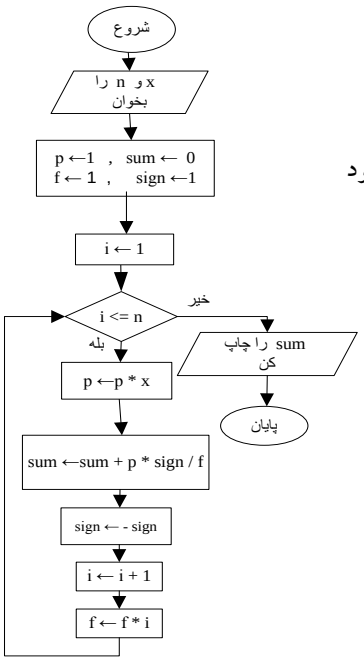
د:1 + i را در i قرار بده

۸. sum را چاپ کن.

۹. پایان

مثال ۱۴ – ۱. الگوریتم و فلوچارتی که x و n را خوانده، مجموع سری زیر را محاسبه می‌کند:

$$sum = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \ldots \pm \frac{x^n}{n!}$$



توضیح: این الگوریتم مانند الگوریتم مثال ۱۳ – ۱ است. با این تفاوت که متغیر دیگری به نام f در نظر گرفته شده است که فاکتوریل شمارنده i را محاسبه می‌کند تا بر p تقسیم شده، در sign ضرب گردد. در داخل حلقه چون مقدار i در f ضرب می‌شود و در f قرار می‌گیرد. در نتیجه f فاکتوریل i خواهد شد.

۱. شروع

۲. x و n را بخوان

۳. یک را در p قرار بده

٤. صفر را در sum قرار بده

۵. یک را در f قرار بده

٦. یک را در sign قرار بده

۷. یک را در i قرار بده

۸. تا وقتی که i <= n تکرار کن

الف: p * x را در p قرار بده

ب: sign * $\frac{p}{f}$ + sum را در sum قرار بده

ج: منفی sign را در sign قرار بده

د:1 + i را در i قرار بده

ه: f * i را در f قرار بده

۹. sum را چاپ کن.

۱۰. پایان

متغیر	هدف
a	ضلع اول مثلث
b	ضلع دوم مثلث
c	ضلع سوم مثلث
s	مساحت یک مثلث
i	شمارنده ۱ تا n
sum	مجموع مساحت مثلث‌ها

مثال۱۵ – ۱. الگوریتم و فلوچارتی که ابتدا عدد صحیح و مثبت n را از ورودی خوانده، سپس، اضلاع n مثلث را از ورودی می‌خواند و مجموع مساحت‌های آن‌ها را محاسبه کرده، چاپ می‌کند.

داده: n (تعداد مثلث‌ها)، a، b و c (اضلاع هر مثلث)

خواسته: مجموع مساحت مثلث‌ها

رابطه : $p = \frac{a+b+c}{2}$

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

$$sum \leftarrow sum + s$$

۱. شروع

۲. n را بخوان

۳. $sum \leftarrow 0$، $i \leftarrow 1$

٤. a، b و c را بخوان

۵. $p \leftarrow \frac{a+b+c}{2}$

٦. $s \leftarrow \sqrt{p(p-a)(p-b)(p-c)}$

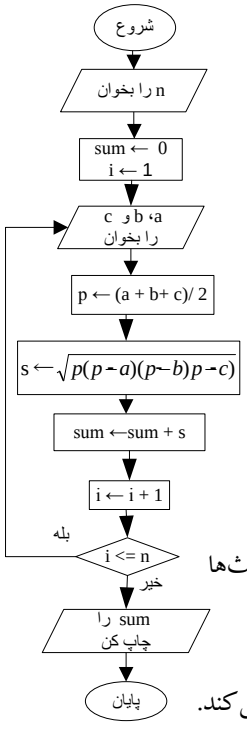
۷. $sum \leftarrow sum + s$

۸. $i \leftarrow i + 1$

۹. اگر $i <= n$ بود برو به مرحله ۴

۱۰. sum را چاپ کن

۱۱. پایان

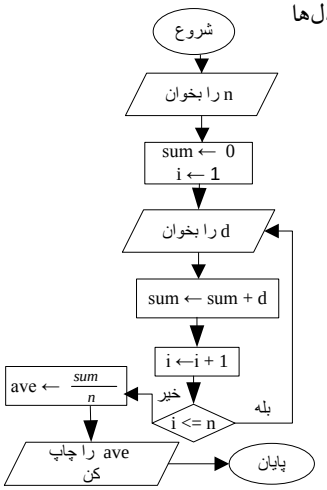


در این الگوریتم ابتدا n تعداد مثلث‌ها را می‌خواند. سپس مجموع مساحت مثلث‌ها (sum) را برابر صفر قرار می‌دهد و مقدار شمارنده حلقه را برابر یک قرار خواهد داد. در داخل حلقه اضلاع یک مثلث را می‌خواند. مساحت مثلث را حساب کرده، با sum جمع می‌کند و سپس به شمارنده یک واحد اضافه می‌کند. اگر شمارنده کمتر یا مساوی n باشد، به ابتدای حلقه برمی‌گردد تا اضلاع مثلث جدیدی را بخواند. در پایان sum را چاپ می‌کند.

مثال۱۶ – ۱. الگوریتم و فلوچارتی که ابتدا عدد صحیح و مثبت n را از ورودی خوانده، سپس معدل n دانشجو را می‌خواند و میانگین معدل آن‌ها را محاسبه کرده و در خروجی می‌نویسد.

داده: n (تعداد معدل)

خواسته: ave (میانگین معدل دانشجویان)



رابطـــــــه

میانگین معدل = مجموع معدل‌ها تقسیم بر تعداد معدل‌ها

مجموع معدل‌ها :sum

شمارنده : i

$i \leftarrow i + 1$

معدل خوانده شده : d

$sum \leftarrow sum + d$

۱. شروع

۲. n را بخوان

۳. $sum \leftarrow 0$، $i \leftarrow 1$

٤. d را بخوان

۵. $sum \leftarrow sum + d$

٦. $i \leftarrow i + 1$

۷. اگر $i <= n$ بود برو به دستور ۴

۸. $ave \leftarrow \frac{sum}{n}$

۹. ave را چاپ کن

۱۰. پایان

این الگوریتم ابتدا تعداد دانشجویان (n) را می‌خواند. سپس، مجموع معدل‌ها (sum) را برابر صفر قرار می‌دهد و یک واحد شمارنده (i) قرار می‌دهد. یک نمره را خوانده، این معدل را با مجموع معدل‌ها جمع کرده در آن قرار می‌دهد. سپس به شمارنده یک واحد اضافه می‌کند. اکنون اگر شمارنده کوچک‌تر یا مساوی تعداد معدل‌ها (n) باشد، معدل بعدی را می‌خواند، وگرنه مجموع معدل‌ها (sum) تقسیم بر تعداد معدل‌ها (n) را در میانگین معدل‌ها (avg) قرار می‌دهد و آن را چاپ می‌کند.

مثال۱۷ – ۱. الگوریتم و فلوچارتی که ابتدا تعداد دانشجویان یک کلاس (عدد صحیح و مثبت n) را از ورودی می‌خواند و سپس، نمرات آن دانشجویان را خوانده و بیشترین و کم‌ترین نمره کلاس را پیدا کرده و در خروجی نمایش می‌دهد (n >= 2) .

تعداد دانشجویان : n ، نمره هر دانشجو: x

کم‌ترین نمره : min ، بیشترین نمره : max

۱. شروع

۲. n را بخوان

۳. $i \leftarrow 2$

٤. x را بخوان

۵. $min \leftarrow x$ و $max \leftarrow x$

٦. x دیگری را بخوان

۷. اگر $x < min$ آنگاه $min \leftarrow x$

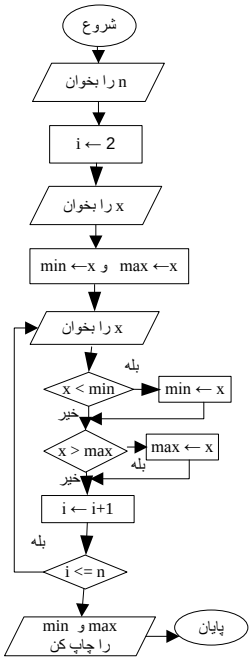
۸. اگر $x > max$ آنگاه $max \leftarrow x$

۹. $i \leftarrow i + 1$

۱۰. اگر $i <= n$ برو به دستور ۶

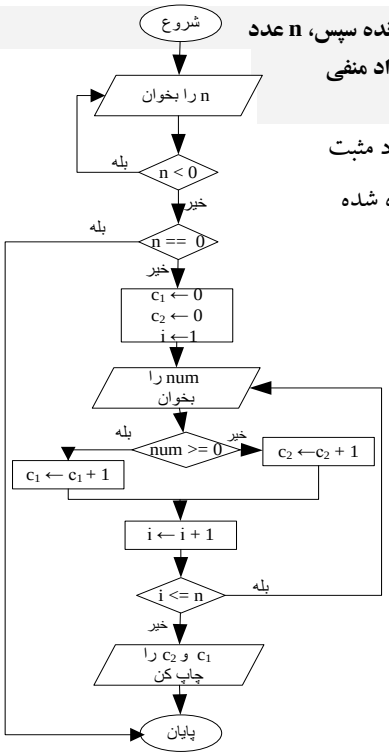
۱۱. min و max را چاپ کن

۱۲. پایان



این الگوریتم ابتدا تعداد دانشجویان را خوانده، سپس در مقدار ۲ را درشمارنده قرار می‌دهد. چون ابتدا یک نمره (x) را خوانده در کم‌ترین نمره (min) و بیشترین نمره (max) قرار می‌دهد. سپس، یک نمره دیگر می‌خواند (دومین نمره را می‌خواند سپس i باید قبل از شروع حلقه ۲ باشد). اگر این نمره خوانده شد کم‌تر از کم‌ترین نمره باشد، آن را در کم‌ترین نمره (min) قرار می‌دهد. سپس نمره را با بیشترین نمره مقایسه می‌کند. اگر این نمره بیشتر از بیشترین نمره مقایسه می‌کند. اگر این نمره بیشتر از بیشترین نمره باشد آن را در بیشترین نمره قرار می‌دهد. در ادامه شمارنده را یک واحد اضافه می‌کند. اگر شمارنده کوچک‌تر یا مساوی n باشد، یک نمره دیگر می‌خواند و این روند را ادامه می‌دهد. وگرنه کم‌ترین و بیشترین نمره را چاپ می‌کند.

مثال ۱۸ – ۱. الگوریتم و فلوچارتی که ابتدا n را از ورودی خوانده سپس، n عدد دیگر را نیز می‌خواند و از بین آن‌ها تعداد اعداد مثبت و تعداد اعداد منفی را تعیین کرده و در خروجی می‌نویسد.



n: تعداد اعدادی را که قرار است خوانده شود، c_1: شمارشگر اعداد مثبت

c_2: شمارشگر اعداد منفی، num : اعداد خوانده شده

i: شمارشگر تعداد اعداد خوانده شده

۱. شروع

۲. n را بخوان

۳. اگر $n < 0$ برو به ۲

٤. اگر $n == 0$ برو به مرحله ۱۱

۵. $c_1 \leftarrow 0$ و $c_2 \leftarrow 0$ و $i \leftarrow 1$

٦. num را بخوان

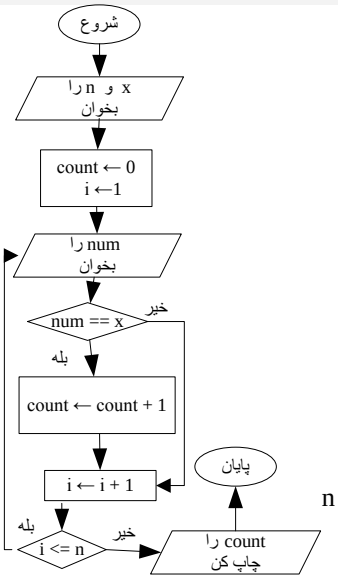
۷. اگر $num >= 0$ آنگاه $c_1 \leftarrow c_1 + 1$

در غیر این صورت $c_2 \leftarrow c_2 + 1$

۸. i ←i + 1
۹. اگر i <= n برو به ۶
۱۰. c_1 و c_2 را چاپ کن
۱۱. پایان

این الگوریتم ابتدا n را می‌خواند. اگر n کوچک‌تر از صفر باشد، عدد دیگری را می‌خواند. اما، اگر n برابر صفر باشد، الگوریتم خاتمه می‌یابد. وگرنه، مقادیر c1(تعداد اعداد مثبت) و c2 (تعداد اعداد منفی) را برابر صفر قرار می‌دهد. چون در ابتدا تعداد اعداد مثبت و منفی باید صفر باشد. سپس، یک را در شمارنده (i) قرار می‌دهد. در ادامه یک عدد (num) را خوانده، اگر عدد بزرگ‌تر یا مساوی صفر باشد، به تعداد اعداد مثبت یک و احد اضافه می‌کند (c1 ← c1 + 1). در پایان به شمارنده یک واحد اضافه می‌کند. اگر i (شمارنده) کوچک‌تر یا مساوی n باشد، عدد بعدی را خواهد خوانده، این روند ادامه می‌یابد. وگرنه، c1 و c2 را چاپ می‌کند.

مثال ۱۹ – ۱. الگوریتم و فلوچارتی که ابتدا اعداد x و n را از ورودی می‌خواند، سپس n عدد دیگر را نیز خوانده، تعداد دفعات تکرار عدد x، در آن n عدد را تعیین کرده و در خروجی چاپ می‌کند.



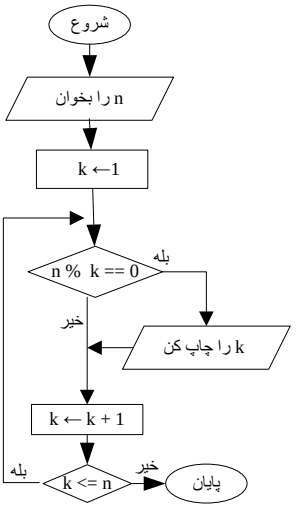
۱. شروع
۲. x و n را بخوان
۳. count ← 0 و i ← 1
۴. num را بخوان
۵. اگر num == x باشد آنگاه count ← count + 1
۶. i ← i + 1
۷. اگر i <= n آنگاه برو به مرحله ۴
۸. count را چاپ کن
۹. پایان

این الگوریتم ابتدا x (عددی که تعداد تکرار آن باید تعیین شود) و n (تعداد اعدادی که باید خوانده شود) را می‌خواند. سپس count (تعداد تکرار) را برابر صفر و i (شمارنده حلقه) را برابر یک قرار می‌دهد. در ادامه عددی را خوانده، در num قرار می‌دهد. اگر x برابر num باشد، به count یک واحد اضافه می‌کند، شمارنده را یک واحد اضافه می‌نماید (i = i + 1).

اکنون اگر i کوچک‌تر یا مساوی n باشد، عددی دیگر را خوانده و این روند را ادامه می‌دهد. وگرنـه، count را چاپ می‌کند.

نکته:	اگر اپراتور % بین دو عدد صحیح قرار گیرد نتیجه باقی مانده تقسیم صحیح خواهد بود.

مثال ۲۰ –۱. الگوریتم و فلوچارتی که ابتدا عدد صحیح و مثبت n را از ورودی می‌خواند، سپس مقسوم علیه-های آن را نمایش می‌دهد:



داده : n

خواسته : مقسوم علیه‌های n

رابطه: اگر n % k == 0 باشد، k مقسوم‌علیه n است

۱. شروع

۲. n را بخوان

۳. k ← 1

٤. اگر n % k == 0 باشد آنگاه k را چاپ کن

۵. k ← k + 1

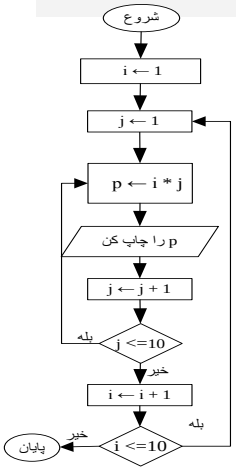
٦. اگر k <= n برو به مرحله ۴

۷. پایان

در این الگوریتم ابتدا n (داده‌ای که باید مقسوم علیه‌های آن نمایش داده شوند) را می‌خواند. سپس یـک را k (شمارنده) قرار می‌دهد. در ادامه،

اگر باقی مانده تقسیم صحیح n بر k برابر صفر باشد (یعنی k مقسوم علیه n است)، k را چاپ می‌کنـد. در پایـان به شمارنده (k) یک واحد اضافه می‌کند و اگر k کوچک‌تر یا مساوی n باشد، این روند ادامه می‌یابـد. وگرنـه، الگوریتم خاتمه خواهد یافت.

مثال ۲۱ – ۱. الگوریتم و فلوچارتی که جدول ضرب ۱۰×۱۰ را در خروجی چاپ می‌کند.



i : شماره سطر (از ۱ تا ۱۰ را می‌شمرد)

j: شماره ستون (برای هر تغییر i از ۱ تا ۱۰ را می‌شمرد)

p: عنصری از جدول ضرب: $p = i \times j$

۱. شروع

۲. i← 1

۳. j← 1

٤. p ← i * j

۵. p را چاپ کن

٦. j ← j + 1

۷. اگر j <= 10 بود برو به مرحله ۴

۸. i ←i + 1

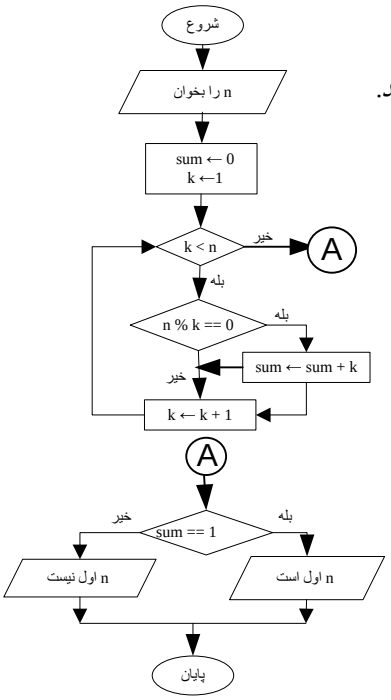
۹. اگر i <= 10 برو به مرحله ۳

۱۰. پایان

مثال ۲۲ – ۱. الگوریتم و فلوچارتی که عدد صحیح و مثبت n را از ورودی می‌خواند، سپس:

الف: تعیین کند n اول است یا خیر. (عددی اول است که مجموع مقسوم علیه‌های آن تا خودش برابر یک شود).

ب: کلیه‌ی اعداد اول کوچک‌تر از n را در خروجی چاپ می‌کند.



پاسخ الف: **داده:** n **خواسته:** n اول است یا خیر

رابطه: اگر n + 1 = مجموع مقسوم‌علیه‌های n آنگاه n اول است

۱. شروع

۲. n را بخوان

۳. k ←1 ، sum←0

٤. تا هنگامی که k < n تکرار کن:

الف: اگر n % k == 0

آنگاه sum ← sum + k

ب: k← k+1

۵. اگر sum == 1 آنگاه بنویس "n اول است"

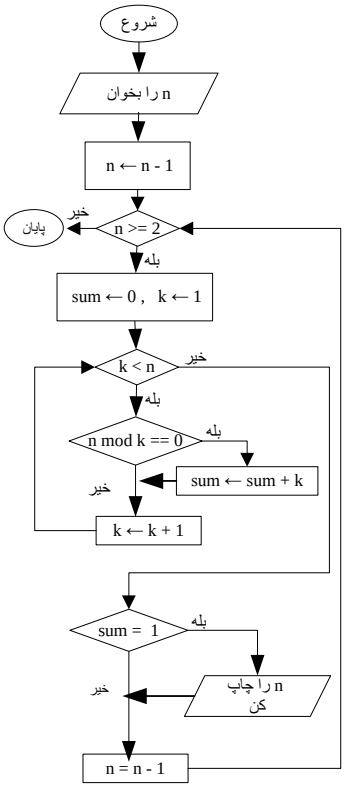
وگرنه، بنویس " n اول نیست"

٦. پایان

این الگوریتم ابتدا n (عددی که باید تعیین شود اول است یا نه) را خوانده، صفر را مجموع مقسوم علیه‌ها (sum) و یک را در شمارنده (k) قرار می‌دهد. تا زمانی که k کوچک‌تر از n باشد، اعمال زیر را انجام می‌دهد:

- اگر باقی‌مانده تقسیم صحیح n بر k صفر باشد، یعنی n بر k بخش‌پذیر است. پس sum + k را در sum قرار می‌دهد.
- به k یک واحد اضافه می‌کند.

زمانی که k بزرگ‌تر یا مساوی n شود، اگر sum برابر یک باشد،آنگاه n اول است را چاپ می کند، وگرنه n اول نیست را نمایش خواهد داد.



پاسخ ب:

۱. شروع

۲. n را بخوان

۳. n ← n -1

۴. تا هنگامی که n >= 2 تکرار کن

الف: k ← 1, sum ← 0

ب: تا هنگامی که k < n تکرار کن

- اگر n % k == 0 باشد

  آنگاه sum ← sum + k

- k ← k + 1

پ: اگر sum == 1 باشد آنگاه n را بنویس

د: n ← n - 1

۵. پایان

این الگوریتم مانند الگوریتم قبلی است. با این تفاوت که به جای این که تعیین کند n اول است، اعداد از n منها یک تا ۲ را با همان الگوریتم تعیین می کند اول است یا نه و اعداد اول را نمایش می‌دهد.

مثال ۲۳ – ۱. الگوریتم و فلوچارتی که ابتدا تعداد دانشجویان یک کلاس را از ورودی می‌خواند(n)، سپس نمرات آن دانشجویان را به ترتیب از ورودی خوانده، نمرات اول و دوم کلاس را تعیین کرده و در خروجی می‌نویسد.

داده تعداد: n

نمرات: x, y

خواسته نمره اول: max1

نمره دوم: max2

k = 1, 2 , ..., n

رابطه: مقایسه

۱. شروع

۲. n را بخوان