

فصل

الگوریتم و فلوجارت



❖ اهداف آموزشی

- ❖ آشنایی با مفاهیم الگوریتم و فلوجارت
- ❖ بررسی ساختار شرطی در فلوجارت
- ❖ بررسی ساختار تکرار در فلوجارت

□ مقدمه

توجه کنید که برای حل هر مسئله و پیاده‌سازی آن با یک زبان برنامه‌نویسی نیاز است یکسری مراحل و فرآیند طی شود. در گام اول باید مسئله به طور کامل آنالیز و مورد تجزیه و تحلیل قرار بگیرد. در گام دوم باید اطلاعات و فرضیات مسئله، استخراج گردد. در گام سوم مراحل حل مسئله تبدیل به فلوچارت و الگوریتم می‌گردد تا هر کاربر یا برنامه‌نویس کامپیوتر با هر زبانی قادر به درک و پیاده‌سازی آن مسئله باشد و در گام آخر برنامه موردنظر از روی فلوچارت توسط یک زبان برنامه‌نویسی کامپیوتر پیاده‌سازی می‌شود.

می‌توان نتیجه بگیریم که الگوریتم و فلوچارت نقش بسیار مهمی در برنامه‌نویسی دارند که به واسطه آنها می‌توان هر مسئله‌ای را با جزئیات کامل به یک برنامه تبدیل کرد. لذا از دانشجویان گرامی تقاضا می‌شود که برای افزایش مهارت خود در کدنویسی حتماً مباحث الگوریتم و فلوچارت را با دقت و توجه بیشتری دنبال کنند.

□ نرم‌افزار^۱

سخت‌افزار کامپیوتر که مجموعه‌ای از قطعات فیزیکی و الکترونیکی می‌باشند به تنهایی قادر نیست که کار خاصی را انجام دهد و هر کاربر برای استفاده از آن نیاز به نرم‌افزار دارد. به عبارتی نرم‌افزار پل ارتباطی بین کاربران و سخت‌افزار کامپیوتر می‌باشد. هر نرم‌افزار کامپیوتر در اصل مجموعه‌ای از برنامه‌ها است که توسط یکسری برنامه‌نویس^۲ کامپیوتر به منظور انجام یک عمل خاص نوشته می‌شوند. هر برنامه نیز مجموعه‌ای از دستورالعمل‌ها و عبارات است که ترتیب انجام آنها یک هدف خاصی را دنبال می‌کند.

-
1. Software
 2. Programmer



□ الگوریتم

مجموعه متناهی از دستورالعمل‌ها که به ترتیب خاصی اجرا شده و مسئله را حل می‌کند، در اصطلاح الگوریتم نامیده می‌شود. به عبارت دیگر الگوریتم یک روش گام به گام برای حل یک مسئله است که دارای چهار ویژگی زیر می‌باشد:

- ۱- ورودی: یک الگوریتم می‌تواند هیچ یا چند ورودی داشته باشد.
- ۲- خروجی: یک الگوریتم باید حداقل یک خروجی داشته باشد.
- ۳- قطعیت: دستورات الگوریتم باید با زبانی دقیق و بی‌ابهام بیان شوند و هر دستورالعمل نیز باید انجام‌پذیر باشد.
- ۴- محدودیت: الگوریتم باید دارای شروع و پایان مشخص باشد به عبارتی طی اجرای مراحل الگوریتم سرانجام خاتمه یابد.

نکته: تفاوت برنامه با الگوریتم در این است که یک الگوریتم حتماً یک خروجی داشته و پایان‌پذیر است در حالی که یک برنامه می‌تواند خروجی نداشته باشد و همچنین پایان‌ناپذیر باشد. برای حل هر مسئله باید ابتدا الگوریتم حل مسأله به دست بیاید. بدین منظور ابتدا باید مسئله با جزئیات کامل و دقیق خوانده شود. سپس فرضیات مسأله استخراج شده و یک ارتباط منطقی بین این فرضیات و روش حل مسئله به وجود بیاید و در مرحله آخر مسئله بر اساس یک زبان برنامه‌نویسی کامپیوتری به صورت یک برنامه پیاده‌سازی گردد.

مثال) خرید دارو از داروخانه:

- ۱- شروع
- ۲- بیمار وارد داروخانه می‌شود.
- ۳- واحد پذیرش داروخانه، نسخه بیمار را جهت تهیه داروی وی از آن تحویل می‌گیرد.
- ۴- اگر دارو یا داروها موجود نباشد به بیمار اطلاع می‌دهد.
- ۵- اگر دارو یا داروها موجود باشد، توسط دکتر داروساز داروها از قفسه‌های مربوطه دارو جمع‌آوری شده و دستورالعمل مصرف هر یک بر روی آنها نوشته می‌شود.
- ۶- بیمار با مراجعه به صندوق با پرداخت هزینه داروها، داروی خود را تحویل می‌گیرد.
- ۷- پایان

همان‌طور که در مثال فوق مشاهده نمودید، بیمار جهت تهیه دارو یکسری مراحل را به ترتیب انجام می‌دهد. این مراحل که از یک نقطه آغاز می‌شود و در یک نقطه پایان می‌پذیرد، در مجموع این الگوریتم چگونگی تهیه دارو از داروخانه نامیده می‌شود.



□ فلوجارت

با توضیحات قبل مشخص شد که جهت حل یک مسأله و پیاده‌سازی آن به یک برنامه کامپیوتری توسط یک زبان برنامه‌نویسی نیاز به یک الگوریتم می‌باشد. ولی با توجه به اینکه الگوریتم‌ها معمولاً به زبان‌های مختلف در دنیا و با نحوه نگارش متفاوت نوشته می‌شوند، یک استاندارد مشخصی برای آنها وجود ندارد. حال اگر برای نمایش مراحل حل یک مسأله از یک نمودار گرافیکی که شامل نمادهای ویژه که هر کدام بیانگر یک عمل خاص هستند، استفاده شود یک استاندارد بین‌المللی برای بیان الگوریتم‌ها به وجود می‌آید که در اصطلاح به آن **فلوجارت** گفته می‌شود. به عبارتی یک فلوجارت همان الگوریتم است ولی در قالب فرم گرافیکی که برای هر کاربر و برنامه‌نویسی با زبان‌های متفاوت قابل فهم می‌باشد.

نکته: با استفاده از فلوجارت به جای الگوریتم فهم مراحل^۱ حل مسئله آسان‌تر شده و برنامه‌نویس به راحتی می‌تواند اشکال زدایی^۲ و ردیابی^۳ در برنامه را انجام دهد.

۱		شروع: برای شروع هر فلوجارت از این نماد استفاده می‌شود.
۲		ورودی: برای خواندن هر مقدار در فلوجارت از این نماد استفاده می‌شود.
۳		شرط: برای نمایش شرطها در فلوجارت از این نماد استفاده می‌شود.
۴		خروجی: برای چاپ یا نمایش هر مقداری در فلوجارت از این نماد استفاده می‌شود.
۵		انتساب و پردازش: برای مقداردهی و محاسبات در فلوجارت از این نماد استفاده می‌شود.
۶		پایان: برای خاتمه دادن به فلوجارت از این نماد استفاده می‌شود.
۷		ارتباط: جهت نمایش ترتیب مراحل مختلف در فلوجارت استفاده می‌شود.
۸		خطوط ارتباطی: جهت نمایش و ارتباط مراحل مختلف در فلوجارت استفاده می‌شود.

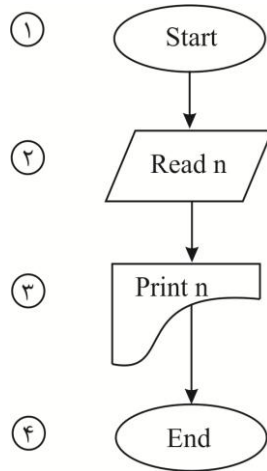
جدول ۱-۲ نمادهای فلوجارت

1. Steps
2. Debugging
3. Tracing



برای درک بهتر و تمرین جهت رسم فلوچارت‌های متفاوت یکسری تمرین را در این فصل بررسی می‌کنیم.

تمرین ۱- فلوچارتی رسم کنید که یک عدد را از ورودی خوانده و آن را در خروجی چاپ کند.



مراحل الگوریتم:

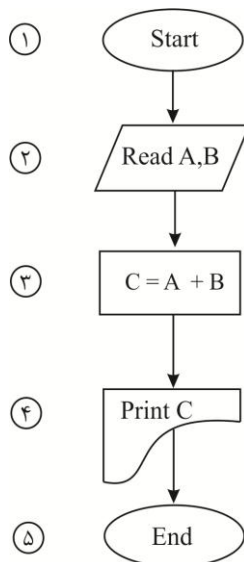
۱- شروع

۲- متغیر n را از ورودی بخوان

۳- متغیر n را در خروجی چاپ کن

۴- پایان

تمرین ۲- فلوچارتی رسم کنید که دو عدد را از ورودی خوانده و حاصل جمع آنها را در خروجی چاپ کند.



مراحل الگوریتم:

۱- شروع

۲- دو متغیر A و B را از ورودی بخوان

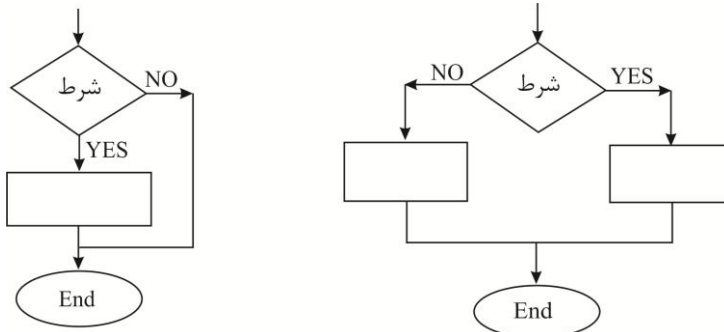
۳- حاصل جمع دو متغیر A و B را در متغیر C قرار بده

۴- متغیر C را چاپ کن

۵- پایان

■ ساختار شرطی در فلوچارت

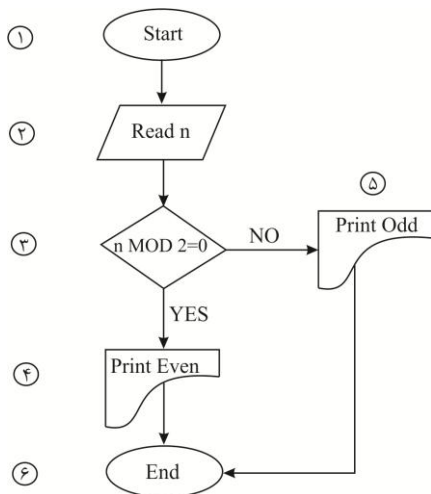
برای پیاده‌سازی ساختار شرطی (IF) در فلوچارت‌ها می‌توان از دو روش پیاده‌سازی استفاده نمود:



چنانچه شرط برقرار باشد شاخه سمت کلمه Yes اجرا می‌گردد، در غیر اینصورت اگر شرط برقرار نباشد شاخه سمت کلمه No مورد پردازش قرار می‌گیرد.

تمرین ۳- فلوچارتی رسم کنید که تشخیص دهد عدد ورودی زوج است یا فرد؟

مراحل الگوریتم:



۱- شروع

۲- متغیر n را از ورودی بخوان

۳- اگر باقیمانده تقسیم بر ۲ متغیر n برابر صفر بود به مرحله ۴ برو در غیر اینصورت به مرحله ۵ برو

۴- پیام زوج بودن متغیر n را چاپ کن و به مرحله ۶ برو

۵- پیام فرد بودن متغیر n را چاپ کن و به مرحله ۶ برو

۶- پایان

■ ساختار تکرار در فلوچارت

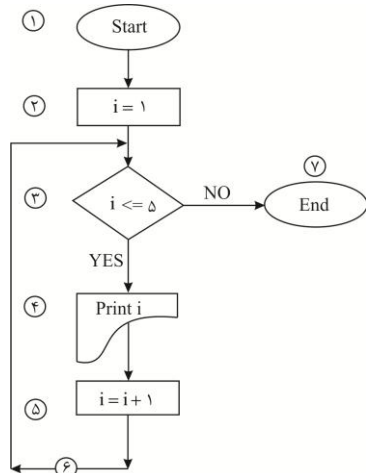
برای پیاده‌سازی ساختار تکرار یا حلقه در فلوچارت‌ها با توجه به اینکه یکسری از دستورات باید بر اساس یک شرط خاص به طور محدود یا نامحدود اجرا گردند، بنابراین لازم است از ساختار شرطی برای اجرای ساختار تکرار استفاده شود.



ساختار کنترلی حلقه می‌تواند بر اساس یک مقدار مشخص، چندین بار تکرار شود که به این نوع حلقه‌ها، **حلقه ثابت** می‌گویند. در هر حلقه نیاز به یک **شمارنده** یا **متغیر حلقه** می‌باشد.

تمرین ۴- فلوچارتی رسم کنید که اعداد ۱ تا ۵ را در خروجی چاپ کند.

مراحل الگوریتم:



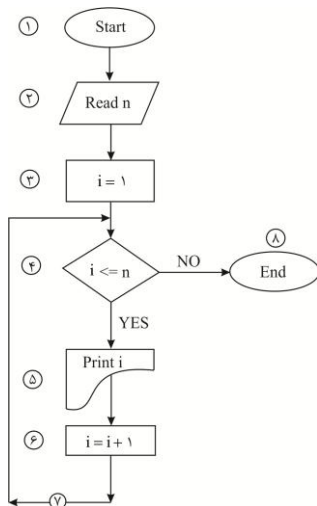
- ۱- شروع
- ۲- مقدار اولیه شمارنده حلقه (i) را برابر ۱ کن
- ۳- اگر مقدار شمارنده حلقه (i) کوچکتر از ۵ بود برو به مرحله ۴، در غیر اینصورت برو مرحله ۷
- ۴- متغیر i را چاپ کن
- ۵- یک واحد به متغیر i اضافه کن
- ۶- برو به مرحله ۳
- ۷- پایان

تذکره: در ساختار تکرار مثال قبل متغیر i نقش شمارنده حلقه را دارد که در شرط حلقه هر بار بررسی می‌شود و تا زمانی که شرط برقرار باشد حلقه تکرار می‌گردد.

در بعضی از ساختارهای تکراری ممکن است تعداد تکرار دستورالعمل مشخص نباشد، به عبارتی تعداد دفعات تکرار نیز به عنوان ورودی به برنامه وارد شود. به این نوع حلقه‌ها، **حلقه متغیر** می‌گویند.

تمرین ۵- فلوچارتی رسم کنید که اعداد ۱ تا n را در خروجی چاپ کند.

مراحل الگوریتم:

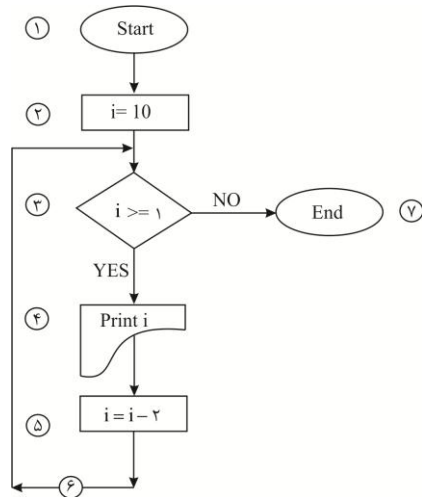


- ۱- شروع
- ۲- متغیر n را از ورودی بخوان
- ۳- مقدار اولیه شمارنده حلقه (i) را برابر ۱ کن
- ۴- اگر مقدار شمارنده حلقه (i) کوچکتر از متغیر n بود برو مرحله ۵، در غیر اینصورت برو مرحله ۸
- ۵- متغیر i را چاپ کن
- ۶- یک واحد به متغیر i اضافه کن
- ۷- برو به مرحله ۴
- ۸- پایان

1. Counter

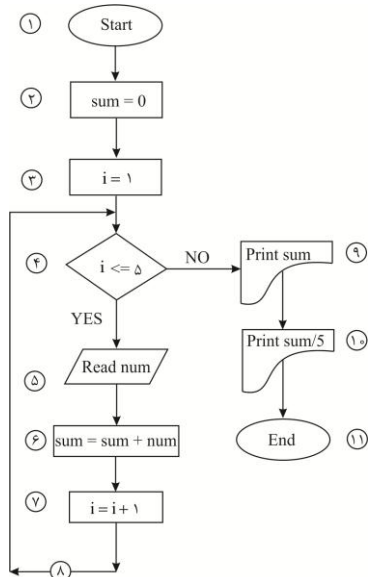
در یکسری از ساختارهای تکرار، حلقه‌ها می‌توانند نزولی باشند. در مثال‌های قبلی، تمامی حلقه‌ها صعودی بودند، یعنی شمارنده حلقه‌ها در هر بار اجرای حلقه اضافه می‌شوند، در حالی که در حلقه‌های نزولی یا کاهنده، از شمارنده حلقه در هر بار اجرای حلقه کاسته می‌شود.

تمرین ۶- فلوجارتی رسم کنید که اعداد زوج بین ۱ تا ۱۰ را به صورت نزولی در خروجی چاپ کند.



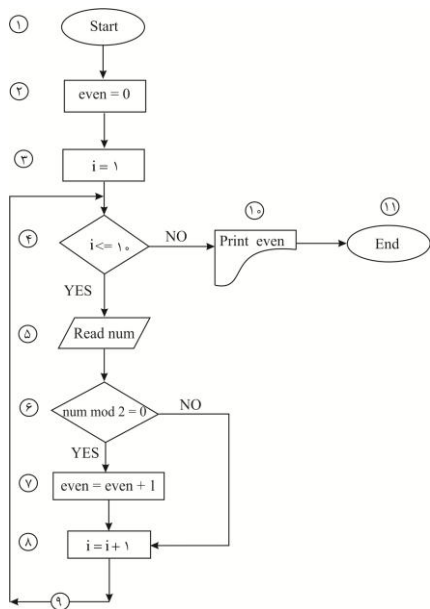
تذکر: از این تمرین به بعد مراحل الگوریتم را دیگر ذکر نمی‌کنیم و فرض را بر این می‌گیریم که دانشجویان گرامی از روی فلوجارت به راحتی می‌توانند مراحل الگوریتم را بنویسند. توجه کنید هدف اصلی این تمرین مهارت پیدا کردن در رسم فلوجارت برای مسائل مختلف است.

تمرین ۷- فلوجارتی رسم کنید که ۵ عدد را از ورودی خوانده و در خروجی حاصل جمع و میانگین آنها را چاپ کند.

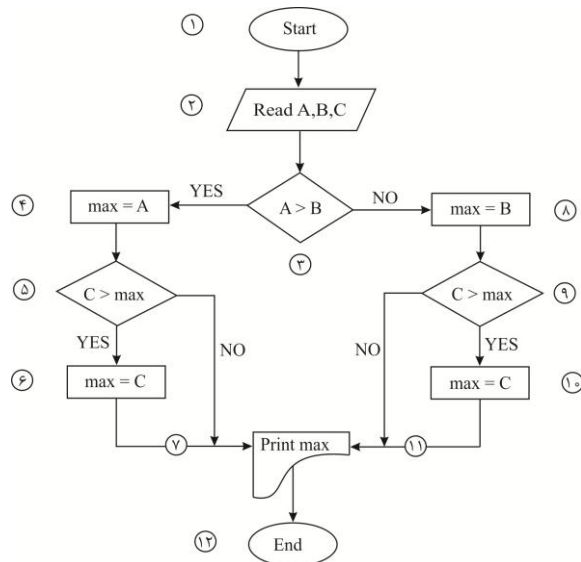




تمرین ۸- فلوجارتی رسم کنید که ۱۰ عدد را از ورودی خوانده و در خروجی تعداد اعداد زوج را چاپ کند.

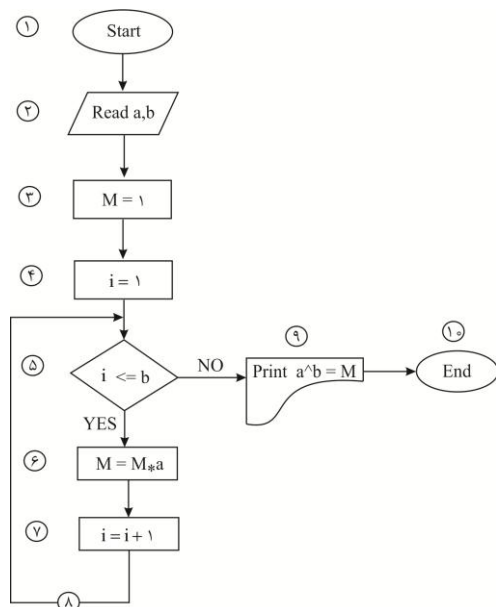


تمرین ۹- فلوجارتی رسم کنید که سه عدد را از ورودی خوانده و در خروجی بزرگترین آنها را چاپ کند.

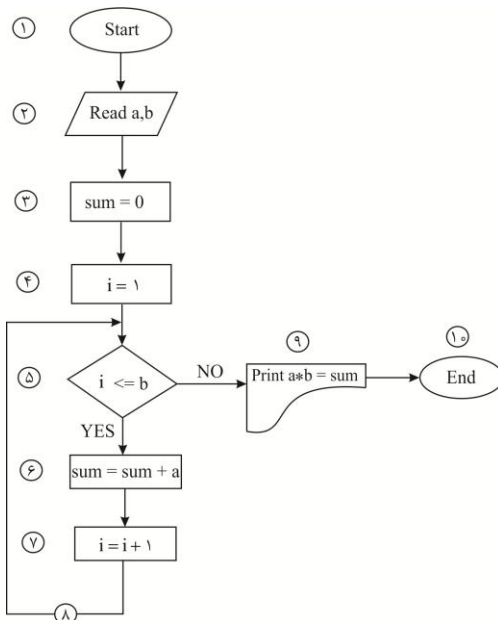




تمرین ۱۰- فلوجارتی رسم کنید که دو متغیر عددی را از ورودی خوانده و سپس متغیر اول را به توان متغیر دوم برساند.

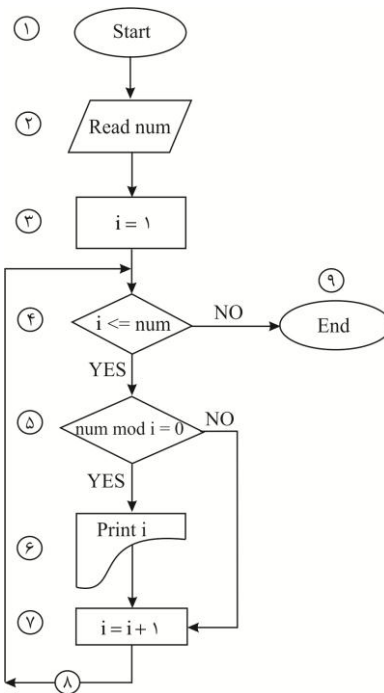


تمرین ۱۱- فلوجارتی رسم کنید که دو متغیر عددی را از ورودی خوانده و سپس متغیر اول را در متغیر دوم ضرب کرده به طوری که از عملگر جمع به جای عملگر ضرب استفاده کند.





تمرین ۱۲- فلوچارتی رسم کنید که یک عدد را از ورودی خوانده و سپس مقسوم‌علیه‌های آن را در خروجی چاپ کند.



تذکر: منظور از مقسوم‌علیه‌های یک عدد تمامی اعداد کوچکتر مساوی خود آن عدد می‌باشد که بر آن بخش‌پذیرند. به طور مثال:

۱۲ → مقسوم‌علیه‌ها {۱, ۲, ۳, ۴, ۶, ۱۲}

۲۴ → مقسوم‌علیه‌ها {۱, ۲, ۳, ۴, ۶, ۸, ۱۲, ۲۴}

۸ خودآزمایی فصل دوم

- ۱- مفاهیم نرم‌افزار، الگوریتم و فلوچارت را تعریف کنید.
- ۲- ویژگی‌های یک الگوریتم را نام برده و توضیح دهید.
- ۳- تفاوت برنامه با الگوریتم را بیان کنید.
- ۴- فرآیند حل یک مسأله را با ذکر یک مثال (خرید دارو از داروخانه) توضیح دهید.
- ۵- فواید استفاده از فلوچارت را بیان کنید.
- ۶- فلوچارتی رسم کنید که n عدد را از ورودی گرفته و در خروجی مجموع و میانگین آنها را چاپ کند.
- ۷- فلوچارتی رسم کنید که n عدد را از ورودی خوانده و در خروجی تعداد اعداد زوج و فرد را چاپ کند.
- ۸- فلوچارتی رسم کنید که اضلاع یک مثلث را از ورودی خوانده و اگر مثلث قائم‌الزاویه بود یک پیام مناسب در خروجی چاپ کند.
- ۹- فلوچارتی رسم کنید که دو عدد را از ورودی خوانده و سپس عدد اول را بر عدد دوم تقسیم صحیح کرده به طوری که به جای عملگر تقسیم از عملگر تفریق استفاده کند و در آخر خارج قسمت تقسیم را در خروجی چاپ کند.
- ۱۰- فلوچارتی رسم کنید که یک عدد از ورودی خوانده اگر عدد برابر یک بود پیام One، اگر عدد برابر دو بود پیام Two، اگر عدد برابر ۳ بود پیام Three و در غیر این صورت پیام Other را در خروجی چاپ کند.

فصل

آشنایی با ساختار زبان C



❖ اهداف آموزشی

- ❖ آشنایی با ساختار برنامه در زبان C
- ❖ آشنایی با نحوه استفاده از دستورات ورودی و خروجی
- ❖ چگونگی استفاده از توضیحات در برنامه‌ها
- ❖ آشنایی با تابع `main()` و عملکرد آن

□ ساختار برنامه در زبان C

تمرین ۱- برنامه‌ای بنویسید که متغیر x را از ورودی خوانده، سپس مقدار تابع $y = 3x - 5$ را در خروجی چاپ کند.

کد برنامه

```
#include <stdio.h>
#include <conio.h>
int main() {
    int x,y;
    printf("please enter the value of X:");
    scanf("%d", &x);
    y=3*x-5;
    printf("\ny=%d", y);
    getch();
    return 0;
}
```

خروجی برنامه

```
please enter the value of X:2
y=1
```

توضیحات برنامه

ساختار یک برنامه در زبان C با تابع اصلی main مشخص می‌گردد. به عبارتی دستورات برنامه در داخل این تابع نوشته می‌شود. به طور معمول در تمامی برنامه‌های نوشته شده به زبان C از