برنامه‌سازی کامپیوتر

جلسه ششم

ساختارهای تصمیم در زبان C

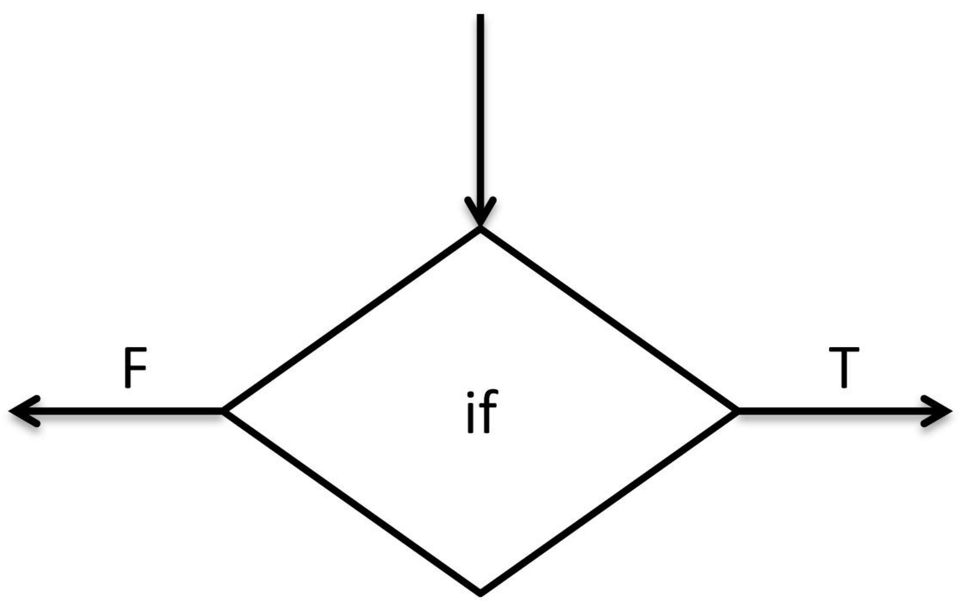
طرح کلی

- ساختار تصمیم if (دستور کنترل شرطی)
- دستورات کنترل غیرشرطی
  - continue
  - break
  - goto
- ساختار تصمیم switch

ساختارهای تصمیم

در مواردی که می‌خواهیم با توجه به شرایطی خاص، مجموعه‌ای از دستورات اجرا شوند و مجموعه‌ای دیگر از دستورات اجرا نشوند از ساختارهای تصمیم استفاده می‌کنیم.

در این ساختارها، شرطی مورد بررسی قرار گرفته و اگر درست باشد تعدادی از دستورات اجرا می‌شوند و در صورتی که شرط نادرست بود تعداد دیگری از دستورات اجرا خواهند شد.



ساختار شرطی if

```
if   (شرط)
{
        دستور ۱ ;
        دستور ۲ ;
        دستور ۳ ;
        ...
        دستور n ;
}
else
{
        دستور ۱ ;
        دستور ۲ ;
        دستور ۳ ;
        ...
        دستور n ;
}
```

- در این ساختار ابتدا شرطی چک می‌شود و در صورتی که ارزش شرط درست باشد، مجموعه‌ای از دستورات اجرا خواهند شد. در صورتی که ارزش شرط نادرست باشد مجموعه‌ای دیگر از دستورات اجرا خواهند شد.
- در صورتی که هر کدام از مجموعه دستورات شامل فقط یک دستور باشد، استفاده از } و { موردنیاز نمی‌باشد.
- در صورتی که ارزش شرط درست باشد دستورات بعد از if اجرا می‌شوند و سپس اجرا از ساختار if خارج می‌شود، اگر ارزش شرط نادرست باشد فقط دستورات بخش else اجرا می‌شود و سپس اجرا از ساختار if خارج می‌شود.
- این ساختار می‌تواند فاقد بخش else نیز باشد.

مثال

برنامه‌ای بنویسید که زوج یا فرد بودن عدد num را چاپ کند.

```
if (num % 2 == 0)
    printf("Zooj");
else
    printf("Fard");
```

برنامه‌ای بنویسید که قدر مطلق عدد n را چاپ کند.

```
if (n < 0)
  n = n * (-1);
printf("%d", n);
```

مثال

برنامه‌ای بنویسید که دو عدد صحیح را از ورودی دریافت کرده و عدد بزرگتر را در خروجی چاپ کند.

متغییرها:

num1 عدد اول

num2 عدد دوم

max عدد بزرگتر

```
#include <stdio.h>
int main()
{
  int  num1, num2, max;
  printf("\nEnter two numbers:");
  scanf("%d %d", &num1, &num2);

  if (num1 > num2)
      max = num1;
  else
      max = num2;

  printf("\nMax of numbers is : %d" , max);
  getch();
  return 0;
}
```

خروجی:

```
Enter two numbers:34 65

Max of numbers is : 65
```

مثال

```
#include <stdio.h>
int main()
{
  float A, B, C, delta, x1, x2;
  printf("Enter A, B, C : ");
  scanf("%f%f%f", &A, &B, &C);
  delta = B*B-4*A*C;
  if (delta >=0)
    {
      x1 = ( -B + sqrt(delta) ) / (2 * A);
      x2 = ( -B - sqrt(delta) ) / (2 * A);
      printf("\nRoots are x1=%6.2f , x2=%6.2f", x1, x2);
    }
  else
    {
      printf("\nReal roots aren't exists.");
    }
  getch();
  return 0;
}
```

برنامه محاسبه ریشه‌های معادله درجه دوم

$Ax^2+Bx+C=0$

```
Enter A, B, C : 1 2 3

Real roots aren't exists.
```

```
Enter A, B, C : 2 8 2

Roots are x1= -0.27 , x2= -3.73
```

ساختار else if

- در استفاده از ساختار if برای بررسی کردن شرط‌های متعدد باید آن‌ها را بصورت تودرتو استفاده کنیم.
- کاربرد if بصورت تودرتو، نه‌تنها موجب طولانی شدن برنامه می‌شود، بلکه از خوانایی برنامه نیز خواهد کاست.
- ساختار else-if می‌تواند جایگزین چندین if تودرتو شود.

```
if   (شرط ۱)
  {
        دستور ۱ ;
          ...
        دستور n ;
  }
else if  (شرط ۲)
  {
        دستور ۱ ;
          ...
        دستور n ;
  }
else if  (شرط ۳)
  {
        دستور ۱ ;
          ...
        دستور n ;
  }
```

مثال

برنامه‌ای بنویسید که نمره عددی دانشجویی را خوانده، معادل حرفی آن را در خروجی چاپ می‌کند.

متغییرها:

grade نمره‌ای که دریافت می‌شود.

17 <= نمره عددی <= 20 → 'A'
15 <= نمره عددی < 17 → 'B'
12 <= نمره عددی < 15 → 'C'
نمره عددی < 12 → 'D'

```
#include <stdio.h>
int main(){
  float grade;
  printf("\nEnter a grade: ");
  scanf("%f", &grade);
  if (grade >= 17 && grade <= 20)
     printf("\ngrade = %5.2f, score=%c", grade, 'A');
  else
     if (grade >= 15 && grade < 17)
          printf("\ngrade = %5.2f, score=%c", grade, 'B');
     else
          if (grade >= 12 && grade <= 15)
               printf("\ngrade = %5.2f, score=%c", grade, 'C');
           else
               if (grade < 12)
                    printf("\ngrade = %5.2f, score=%c", grade, 'D');
  getch();
  return 0;
}
```

```
#include <stdio.h>
int main(){
  float grade;
  printf("\nEnter a grade: ");
  scanf("%f", &grade);
  if (grade >= 17 && grade <= 20)
      printf("\ngrade = %5.2f, score=%c", grade, 'A');
  else  if (grade >= 15 && grade < 17)
       printf("\ngrade = %5.2f, score=%c", grade, 'B');
  else  if (grade >= 12 && grade <= 15)
       printf("\ngrade = %5.2f, score=%c", grade, 'C');
  else  if (grade < 12)
       printf ("\ngrade = %5.2f, score=%c", grade, 'D');
  getch();
  return 0;
}
```

مثال

برنامه‌ای بنویسید که روز هفته را گرفته و نام آن را چاپ کند.

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter number of day between 1 – 7: ");
  scanf("%d", &n);
  if (n ==1)  printf("\n Saturday");
  if (n ==2)  printf("\n Sunday");
  if (n ==3)  printf("\n Monday");
  if (n ==4)  printf("\n Tuesday");
  if (n ==5)  printf("\n Wednesday");
  if (n ==6)  printf("\n Thursday");
  if (n ==7)  printf("\n Friday");
  getch();
  return 0;
}
```

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter number of day between 1 - 7: ");
  scanf("%d", &n);
  if (n ==1)  printf("\n Saturday");
  else if (n ==2)  printf("\n Sunday");
  else if (n ==3)  printf("\n Monday");
  else if (n ==4)  printf("\n Tuesday");
  else if (n ==5)  printf("\n Wednesday");
  else if (n ==6)  printf("\n Thursday");
  else if (n ==7)  printf("\n Friday");
  else printf("\n Error");
  getch();
  return 0;
}
```

انتقال کنترل غیرشرطی

- دستور کنترل شرطی if بعد از بررسی کردن شرط، با توجه به نتیجه شرط اقدام به اجرای دستورالعمل‌ها می‌کند. برخی دستورات دیگر بدون بررسی کردن هیچ شرطی قادر به انتقال کنترل برنامه به از نقطه‌ای به نقطه دیگر هستند. (break; continue; goto;)
- دستور **break** موجب خروج از حلقه‌های تکرار می‌شود. نحوه استفاده از این دستور به صورت زیر می‌باشد.

break ;

- اگر چندین حلقه تودرتو وجود داشته باشد، این دستور موجب خروج از داخلی‌ترین حلقه می‌شود.
- برای خاتمه دادن به ساختار switch نیز از این دستور استفاده می‌شود.

مثال

برنامه‌ای بنویسید تعدادی عدد را از ورودی خوانده تعداد اعداد زوج و فرد را مشخص می‌کند و به خروجی می‌برد. آخرین عدد ورودی، صفر است.

متغییرها:

num عددی که دریافت می‌شود.

count تعداد اعداد زوج وارد شده.

n تعداد کل اعداد وارد شده.

```
#include <stdio.h>
int main()
{
  int num, count = 0, n = 0;
  while (1) {
        printf("\nEnter a number: ");
        scanf("%d", &num);
        if (num = = 0)
           break;
        n ++;
        if (num % 2 = = 0)
             count ++;
  }
  printf("\nevents = %d, odds=%d", count, n - count);

  getch();
  return 0;
}
```

خروجی:

```
Enter a number: 12
Enter a number: 13
Enter a number: 21
Enter a number: 15
Enter a number: 5
Enter a number: 3
Enter a number: 4
Enter a number: 8
Enter a number: 0
events = 3, odds=5
```

continue

- دستور**continue** در حلقه تکرار موجب انتقال کنترل به ابتدای حلقه می‌شود. پس از انتقال کنترل به ابتدای حلقه، شرط حلقه مورد بررسی قرار می‌گیرد، چنانچه شرط درست باشد، اجرای دستورات حلقه ادامه می‌یابد وگرنه حلقه تکرار خاتمه می‌یابد.
- بصورت زیر مورد استفاده قرار می‌گیرد.

```
continue ;
```

مثال

برنامه‌ای بنویسید که تمامی اعداد مابین ۱۰ و ۵۰ را بجز اعداد مضرب ۷ بصورت نزولی چاپ کند.

```
#include <stdio.h>

int main()
{
  int i;
  for (i=50; i >=10 ; i--)
     {
       if (i % 7 == 0)
            continue;
       else
            printf("\n%d", i);
     }
  getch();
  return 0;
}
```

goto

- دستور **goto** سبب انتقال کنترل از نقطه‌ای به نقطه دیگر از برنامه می‌شود.
- این دستور معمولاً به ندرت مورد استفاده قرار می‌گیرد. چون استفاده از آن خوانایی برنامه را بسیار کاهش می‌دهد.
- نحوه استفاده از این دستور به‌صورت زیر می‌باشد.

goto <برچسب> ;

- برچسب دستور همانند متغییرها نامگذاری می‌شود و به کولن (:) ختم می‌شود. انتقال کنترل توسط goto فقط داخل یک تابع امکان‌پذیر است.

برچسب :	int i = 0 ;
دستور ۱ ;	lable1:
دستور ۲ ;	printf("%d\n" , i);
دستور ۲ ;	i ++ ;
. . . .	if (i < 10)
دستور n ;	goto lable1;
goto برچسب ;	printf("finish") ;

مثال

برنامه‌ای که با استفاده از دستور goto حلقه تکراری را ایجاد می‌کند.

```
#include <stdio.h>
#include <conio.h>

int main()
{
  int x = 1;
  clrscr();
loop1:
  x ++;
  if (x < 100)
     goto loop1;
  printf("\nThe maximum value of x is : %d", x );
  getch();
  return 0;
}
```

خروجی:

```
The maximum value of x is : 100
```

ساختار تصمیم switch

```
switch   (عبارت) {
        case  <مقدار ۱> :
              <دستورات ۱>
              break ;
        case  <مقدار ۲> :
              <دستورات ۲>
              break ;
              . . .
              . . .
              . . .
        default :
              <دستورات n>
}
```

- از این ساختار برای تصمیم‌گیری‌های چندگانه براساس مقادیر مختلف یک عبارت استفاده می‌شود.
- معمولاً در تمام تصمیم‌گیری‌هایی که بیش از سه انتخاب وجود داشته باشد، بهتر است از ساختار switch استفاده شود.
- این ساختار بصورت زیر مورد استفاده قرار می‌گیرد:
  - ابتدا عبارت مقابل switch به مقدار صحیح ارزیابی می‌شود و مقدار آن تعیین می‌شود.
  - اگر مقدار عبارت برابر با <مقدار ۱> باشد <دستورات ۱> اجرا می‌شوند و اجرای دستورات تا رسیدن به break ادامه خواهد یافت.
  - دستور break اجرای برنامه را از ساختار switch خارج می‌سازد.
  - در صورتی که مقدار عبارت با <مقدار ۱> برابر نباشد، با <مقدار۲> مقایسه می‌شود و همین روند ادامه می‌یابد.
  - تا زمانی که مقدار عبارت برابر با یکی از مقادیر نباشد عمل مقایسه ادامه می‌یابد.
  - اگر مقدار عبارت با هیچ کدام از مقادیر برابر نباشد، دستورات بخش default به اجرا درمی‌آید.

مثال

برنامه‌ای بنویسید که روز هفته را گرفته و نام آن را چاپ کند.

```
#include <stdio.h>
int main()
{
  int n;
  printf("Enter number of day between 1 – 7: ");
  scanf("%d", &n);
  if (n ==1)  printf("\n Saturday");
  if (n ==2)  printf("\n Sunday");
  if (n ==3)  printf("\n Monday");
  if (n ==4)  printf("\n Tuesday");
  if (n ==5)  printf("\n Wednesday");
  if (n ==6)  printf("\n Thursday");
  if (n ==7)  printf("\n Friday");
  getch();
  return 0;
}
```

```
#include <stdio.h>
int main()
{  int n;
  printf("Enter number of day between 1 - 7: ");
  scanf("%d", &n);
  switch (n)  {
      case 1:  printf("\n Saturday"); break;
      case 2:  printf("\n Sunday"); break;
      case 3:  printf("\n Monday"); break;
      case 4:  printf("\n Tuesday"); break;
      case 5:  printf("\n Wednesday"); break;
      case 6:  printf("\n Thursday"); break;
      case 7:  printf("\n Friday"); break;

      default : printf("\n Error");        }
  getch();
  return 0;
}
```

نکته

- ساختار switch می‌تواند فاقد بخش default باشد. در این صورت اگر مقدارعبارت با هیچ یک از مقادیر برابر نباشد، هیچ کدام از دستورات داخل switch اجرا نخواهد شد.
- مقادیر موجود در caseها نمی‌توانند باهمدیگر مساوی باشند. یعنی هیچ کدام از مقادیر <مقدار۱> ، <مقدار۲> و ... نباید مساوی باشند.
- اگر در یک case از دستور break استفاده نشود، با مقدار case بعدی or می‌شود. برای اینکه دو یا چند شرط را در ساختار switch باهمدیگر or کنیم، آن‌ها را بدون break پشت سرهم می‌نویسیم.
- یکی از تفاوت‌های if و switch در این است که در ساختار if می‌توان عبارات منطقی یا رابطه‌ای را مورد بررسی قرار داد ولی در ساختار switch فقط مساوی بودن مقادیر مورد بررسی قرار می‌گیرد.
- چند ساختار switch را می‌توان به صورت تودرتو مورد استفاده قرار داد.

تمرینات تکمیلی

- برنامه‌های زیر را به ترتیب با استفاده از یکی از کامپایلرهای زبان C طراحی، پیاده‌سازی و اجرا کنید.

1. برنامه‌ای بنویسید که سه عدد صحیح را از ورودی گرفته و بزرگترین عدد را در خروجی نمایش دهد.
2. برنامه‌ای بنویسید که عددی را گرفته و مقسوم علیه‌های آن را نمایش دهد.
3. برنامه‌ای بنویسید که عددی صحیح را دریافت کرده و نمایش دهد که آن عدد اول است یا نه.
4. برنامه‌ای بنویسید که تعدادی عدد را به ترتیب دریافت کرده و مقسوم علیه‌های آن‌ها را نمایش دهد. وقتی عدد صفر وارد شد، برنامه به پایان می‌رسد.
5. برنامه‌ای بنویسید که با استفاده از ساختار switch شماره ماه شمسی را گرفته و نام آن ماه را نمایش دهد. (آبان → ۸)

جمع‌بندی

- ساختار تصمیم if (دستور کنترل شرطی)
- دستورات کنترل غیرشرطی
  - continue
  - break
  - goto
- ساختار تصمیم switch