C - Programming

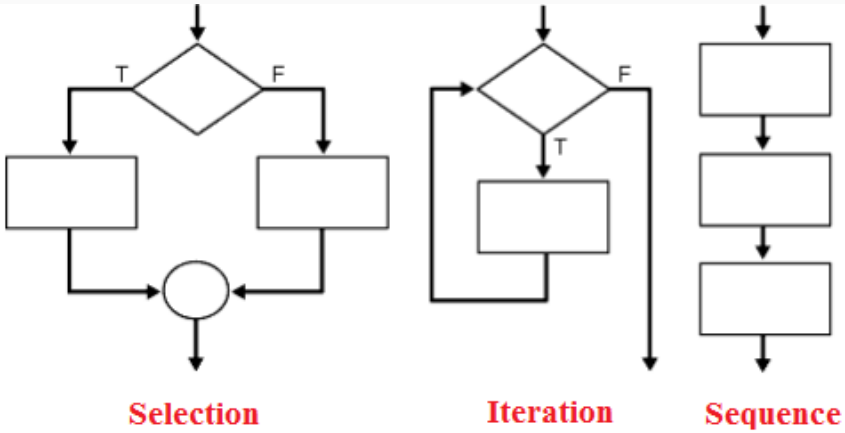
Conditional Statements

Jayakumar P

Department of Computer Science and Applications

Flow of control

- *Flow of control* or *control flow*, refers to the order in which the statements in a program are executed
- Three types of control flow
    1. Sequential - (Top down, default)
    2. Conditional - (also called selection, If else)
    3. Repetitive - (also called Iteration, Loops)



Sequential Control Flow

- In Sequential control flow the statements are executed in a top down manner.
- Each statement is executed only after the previous statement is executed completely.
- Default control flow mechanism

```c
/*Program to find average*/
#include<stdio.h>
int main()
{
    int a,b,c,s;
    float avg;
    scanf("%d%d%d",&a,&b,&c);
    s=a+b+c;
    avg=s/(float)3;
    printf("%f",avg);
    return 0;
}
```

Conditional Control Flow

- Syntax of if statement
  if (*test expression*)
  {
    statements...
  }
- For example, following conditional statement checks whether x (given by the user) is a positive number

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if (x>0)
        printf("Positive");
    return 0;
}
```

Example 1

Write a program to check whether a number given by user is even.

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if (x%2==0)
        printf("Even Number");
    return 0;
}
```

if else statement

The syntax of if else statement

```
if (test-expression)
{
  statements...
}
else
{
  statements...
}
```

For example, the following program checks whether a number given by user is positive or non-positive

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if (x>0)
        printf("Positive");
    else
        printf("Non-positive");
    return 0;
}
```

Example 2

Write a program to check whether a number given by user is even or odd.

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if (x%2==0)
        printf("Even Number");
    else
        printf("Odd Number");
    return 0;
}
```

Q1

Q1. Write a C program to read a number and check whether it is an even number greater than 5.

Q1 - Solution

Q1. Write a C program to read a number and check whether it is an even number greater than 5.

Soln.

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if ( x % 2 == 0 ) && ( x > 5 )
        print("Yes");
    else
        print("No");
    return 0;
}
```

if else if statement

Used to check a set of conditions in sequence. Syntax of if else if is,

```
if (test-expression)
{
statements...
}
else if (test-expression)
{
statements...
}
else
{
statements...
}
```

The final else block may be omitted.

Example

The following program checks whether a given number is positive, or negative, or zero.

```c
#include<stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if (x>0)
        printf("Positive");
    else if (x<0)
        printf("Negative");
    else
        printf("Zero");
    return 0;
}
```

Q2

Q2. Write a C program to read a number between 1 and 7 and print the corresponding week day. For example, if the user input is 1, then output Monday, if the input is 2, then print Tuesday etc.

Q2 - Solution

```
1       scanf("%d",&x);
2       if (x == 1)
3           printf("Monday");
4       else if (x == 2)
5           printf("Tuesday");
6       else if (x == 3)
7           printf("Wednesday");
8       else if (x == 4)
9           printf("Thursday")
10      else if (x == 5)
11          printf("Friday");
12      else if (x == 6)
13          printf("Saturday");
14      else if (x == 7)
15          printf("Sunday");
16      else
17          printf("Invalid Number")
```

Q3

Q3. Find the error in the following C program that reads the total marks of student and decides the corresponding grade.

```
1  scanf("%d",&marks);
2  if (marks > 90)
3      printf("A Grade");
4  if (marks > 80)
5      printf("B Grade");
6  if (marks > 70)
7      printf("C Grade");
8  if (marks > 60)
9      printf("D Grade");
10 else
11     printf("F Grade");
```

Q3- Solution

```
scanf("%d",&marks);
if (marks > 90)
    printf("A Grade");
else if (marks > 80)
    printf("B Grade");
else if (marks > 70)
    printf("C Grade");
else if (marks > 60)
    printf("D Grade");
else
    printf("F Grade");
```

Q4

Alex has to attend a lectures and b practicals tomorrow. He wonders whether he can take enough writing implements. He writes lectures using only pens and he can write down c lectures using one pen. For practical classes he uses only pencils and he can draw d practicals using a pencil.

Alex's pencil-case can hold no more than k writing implements. Now he wants to know how many pens and pencils should he take. Help him to determine it, or tell that his pencilcase doesn't have enough room for all the implements he needs tomorrow! No need to minimize the number of writing equipment.

Q4-Solution

if $ceil(a/c) + ceil(b, d) \leq k$ then print Yes, else print NO.

Ref: https://codeforces.com/problemset/problem/1244/A

Nested if else

C language allows nesting of if else statements.
Example. Following program finds largest among three numbers given by the user.

```
int main(){
int a, b, c;
scanf("%d %d %d", &a, &b, &c);
if(a > b)
    if(a > c)
        printf("%d", a);
    else
        printf("%d", c);
else
    if(b > c)
        printf("%d", b);
    else
        printf("%d", c);
return 0;
```

Q6

Write a C program to print three numbers given by the user in the increasing order.

Q6 - Solution

```
1  if (( x < y ) && ( x < z ) )
2      if ( y < z )
3          min = x, mid = y, max = z;
4      else
5          min = x, mid = z, max = y;
6  else if ( y < z ) &&  ( y < x )
7      if ( x < z )
8          min = y, mid = x, max = z;
9      else
10         min = y, mid = z, max = x;
11 else
12     if ( x < y )
13         min = z, mid = x, max = y;
14     else
15         min = z, mid = y, max = x;
16 printf("Ascending order %d %d %d ", min, mid, max)
```

The dangling else problem

Consider the following program

```c
#include <stdio.h>

int main()
{
    int x;
    scanf("%d",&x);
    if (x>5)
        if (x<10)
            printf("Number is greater than 5 and less than 10");
    else
        printf("Number is less than 5");

    return 0;
}
```

input

```
4


...Program finished with exit code 0
Press ENTER to exit console.
```

The else block is automatically paired with the closest if block (the if block just above the else)

- This is called dangling else problem.
- solution - specify the blocks clearly using brackets.

```c
#include <stdio.h>

int main()
{
    int x;
    scanf("%d",&x);
    if (x>5)
    {
        if (x<10)
            printf("Number is greater than 5 and less than 10");
    }
    else
        printf("Number is less than 5");

    return 0;
}
```

input

```
4
Number is less than 5

...Program finished with exit code 0
```

Switch Statement

- Switch statement is used to perform multi way selection.
- Syntax is,

```
switch(expr)
{
    case constant1: stmtList1;
    break;
    case constant2: stmtList2;
    break;
    case constant3: stmtList3;
    break;
    ..............................
    ..............................
    default: stmtListn;
    break;
}
```

- Switch statement evaluates the expression and switches the control flow to the matching case.
- If none of the case matches then default statement is executed
- The case constants must be integer or character constants.
- Switch statement can test for equality only. (unlike if else if ladder)

Switch Example

```
1       scanf("%d",&x);
2       switch (x)
3       {
4        case 1:  printf("Monday"); break;
5        case 2:  printf("Tuesday"); break;
6        case 3:  printf("Wednesday"); break;
7        case 4:  printf("Thursday"); break;
8        case 5:  printf("Friday"); break;
9        case 6:  printf("Saturday"); break;
10       case 7:  printf("Sunday"); break;
11       default:  printf("Invalid"); break;
12      }
```

Q7

Write a program that checks whether a character entered by the user is a vowel or not (using switch).

Q7- Solution

```c
scanf("%c", &c);
switch(c){
case 'a': case 'A':
case 'e': case 'E':
case 'i': case 'I':
case 'o': case 'O':
case 'u': case 'U':
printf("%c is a vowel!\n", c);
break;
default:
printf("%c is not a vowel!\n", c);
break;
}
```