Programming Using C
Week 03-2

Decision Making Statements-if if else nested if

Prepared By: REC Faculty 4.0 Team

TOPICS

- ✔ INTRODUCTION- SEQUENCE STRUCTURE
- ✔ NEED OF SELECTION STRUCTURE
- ✔ TYPES OF SELECTION STRUCTURE
- ✔ IF STATEMENT SYNTAX & EXAMPLES
- ✔ IF ELSE STATEMENT SYNTAX & EXAMPLES
- ✔ NESTED IF STATEMENT SYNTAX & EXAMPLES
- ✔ MCQ

INTRODUCTION-PROGRAM CONTROL

- The Execution of a Program begins from the main() function. Statements within the main() function are then executed from top-down style, line-by-line.
- However, the order of the execution within the main() body may be branched.
- Changing the order in which statements are executed is called program control.

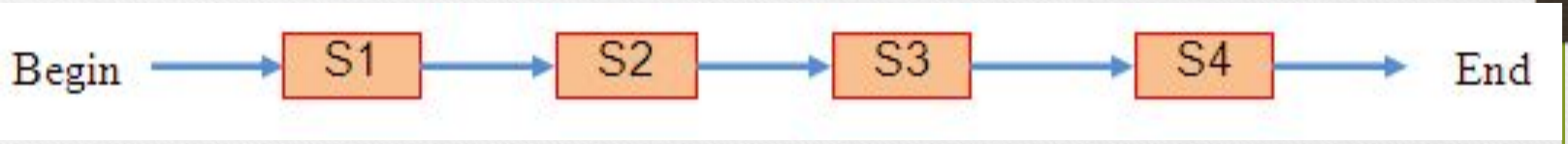
INTRODUCTION-PROGRAM CONTROL STRUCUTRE

There are three types of program control structure in C:

- **Sequence** control structure.
- **Selection structures**
- **Repetition (loop)** structure.
- Certain C functions and keywords also can be used to control the program flows.

SEQUENCE CONTROL STRUCTURE

float paidRate = 5.0, sumPaid, paidHours = 25;	S1
sumPaid = paidHours * paidRate;	S2
printf("Paid sum = $%.2f \n", sumPaid);	S3
return 0;	S4



- One entry point and one exit point.
- Conceptually, a control structure like this means a sequence execution

SELECTION STRUCTURE

Need: There come situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next.

Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the specific block of code.

- Selection Structure in programming languages decides the direction of flow of program execution.
- **Selection Structure is also known as Decision Making Statements**

TYPES OF SELECTION STRUCTURE

- if statement
- if else statement
- nested if statement
- if else if ladder statement
- switch statement

SELECTION STRUCTURE:IF STATEMENT

`if (condition)`	`if (condition)`
`statement;`	`{ statements;}`
`next_statement;`	`next_statement;`

1. `(condition)` is evaluated.
2. If `TRUE` (non-zero) the `statement` is executed.
3. If `FALSE` (zero) the `next_statement` following the `if` statement block is executed.
4. So, during the execution, based on some condition, some codes were skipped.

EXAMPLE 1

```
#include <stdio.h>
int main()
{
    int i = 10;
    if (i > 15)
    {
       printf("10 is less than 15");
    }
      printf("I am Not in if");
 return 0;
}
```

O/P: I am Not in if

SELECTION STRUCTURE:IF-ELSE STATEMENT

`if (condition)`	`if (condition)`
`statement_1;`	`{ a block of statements;}`
`else`	`else`
`statement_2;`	`{ a block of statements;}`
`next_statement;`	`next_statement;`

Explanation:

1. The `(condition)` is evaluated.
2. If it evaluates to non-zero (TRUE), `statement_1` is executed, otherwise, if it evaluates to zero (FALSE), `statement_2` is executed.
3. They are mutually exclusive, meaning, either `statement_1` is executed or `statement_2`, but not both.
4. `statements_1` and `statements_2` can be a block of codes and must be put in curly braces.

EXAMPLE

```
#include <stdio.h>
  int main()
  {
    int i = 20;
    if (i < 15)
        printf("i is smaller than 15");
    else
        printf("i is greater than 15");
    return 0;
}
```

Output: i is greater than 15

SELECTION STRUCTURE: NESTED IF-ELSE STATEMENT

- We can write an entire if else construct within the body of an if statement or else statement.
- This known as nested if statements.

SELECTION STRUCTURE: NESTED IF-ELSE STATEMENT

```
if(test-condition-1)
{
        if(test-condition-2)
        {
                statement-1;
        }
        else
        {
                statement-2;
        }
}
else
{
        statement-3;
}
```

SELECTION STRUCTURE: NESTED IF-ELSE STATEMENT

In this nested form, condition_1 is evaluated. If it is FALSE, statement_3 is executed and the entire nested if statement is terminated.
If TRUE, control goes to the second if (within the first if) and condition_2 is evaluated.
If it is zero FALSE, statement_2 is executed; and the entire nested if statement is terminated.
If TRUE, statement_1 is executed; and the entire nested if statement is terminated.
Again, only one of the statements is executed other will be skipped.
If the else is used together with if, always match an else with the nearest if before the else.

SELECTION STRUCTURE: NESTED IF-ELSE STATEMENT

If the, if else statements is not properly braced in nested if, it ends up in dangling else problem.

The dangling else problem is solved by two ways:

- Implicitly by the compiler (by matching the else clause with the las occurring unmatched if).
- Explicitly by the user (using braces).

EXAMPLE 1

```
#include <stdio.h>
int main()
{
        int a, b, c;
        printf("Enter three numbers : ");
        scanf("%d %d %d", &a, &b, &c);
        if(a > b)
        {
                if(a > c)
                        printf("%d is the biggest number", a);
                else
                        printf("%d is the biggest number", c);
        }
        else
        {
                if(b > c)
                        printf("%d is the biggest number", b);
                else
                        printf("%d is the biggest number", c);
        }
        return 0;
}
```

MCQS

```
#include <stdio.h>
int main()
{
    int i;
    if (printf("0"))
        i = 3;
    else
        i = 5;
    printf("%d", i);
    return 0;
}
```

Predict the output of above program?

A.3

B.5

C.03

D.05

Answer:C

```
#include <stdio.h>
int main()
{
int x = 40, y = 30, z = 80;
if (x < y < z)
printf("REC");
else
printf("RIT");
return 0;
}
```

A. REC
B. RIT
C. Compile time error
D. None of these

ANSWER: A

What is the output of the following C code?

```
#include <stdio.h>
int main()
{
if (0) { printf("First statement"); }
if (-10) { printf("Second statement"); }
if (80 - 10 * 8) { printf("Third statement"); }
return 0;
}
```

A. Only First statement is printed

B. Only Second statement is printed

C. Both Second statement and Third statement are printed

D. No lines are printed

ANSWER: B

Find the output of the following program.

```
#include<stdio.h>
 int main()
{
int x = 5;
if (x >= 6);
printf("hello");
return 0;
}
```

A. It will display hello
B. It will throw an error
C. Nothing will be displayed
D. Compiler dependent

ANSWER: A

What will be the output?

```
#include <stdio.h>
int main()
{
if (1)
    printf("1");
printf("2");
else
 printf("3");
printf("4");
return 0;
}
```

A. 1 2 3 4

B.1 2

C.Compilation error

D. 3 4

ANSWER: C

What will be the output?

```
#include <stdio.h>
int main()
{
int x = -0.5;
if (!x)
printf("hello"); return 0;
}
```

A. hello
B. Compilation error
C. Nothing will be displayed
D. Compiler dependent

ANSWER: A

```c
#include <stdio.h>
int main()
{
    int i;
    if (i = 0, 2, 3)
        printf("REC ");
    else
        printf("Programming Logic ");
    printf("%d\n", i);
    return 0;
}
```

A. REC0
B. Compilation error
C. Programming Logic3
D. Programming Logic0

ANSWER: A

What will be the output?

```
#include <stdio.h>
int main()
{
        int x = 2;
        if (x = 1)
                printf("TRUE");
        else
                printf("FALSE");
        return 0;
}
```

What will be the output of given program?

```
#include <stdio.h>
int main()
{
    int i = 1, j = -1;
    if ((printf("%d ", i)) < (printf("%d ", j)))
        printf("%d ", i);
    else
        printf("%d ", j);
    return 0;
}
```

1 -1 1

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    int i = 0, j = 1, k = 0;
    if (++k, j, i++)
        printf("%d %d %d", i, j, k);
    return 0;
}
```

NO output

Problem 1

Lily likes to play games with integers. She has created a new game where she determines the difference between a number and its reverse. For instance, given the number 12, its reverse is 21. Their difference is 9.Beautiful numbers are defined as numbers where |i – reverse(i)| is divisible by k. If a day's value is a beautiful number, it is a beautiful day. Print whether the given number is beautiful or not.

Input Format

A single line of two space-separated integers describing the respective values of i, and k.

Constraints:

1 <= i, k<= 99

Output Format:

Print Yes if the given number is beautiful, otherwise No.

Sample Input and Output:

Input:

20 6

Output :

Yes

Solution

```c
#include<stdio.h>
int main()
{
    int i,k,a,b,c;
    scanf("%d%d",&i,&k);
    a=i%10;
    b=i/10;
    c=a*10+b;
    ((i-c)%k==0)?printf("Yes"):printf("No");
    return 0;

}
```

Problem 2

Two cats and a mouse are at various positions on a line. You will be given their starting positions. Your task is to determine which cat will reach the mouse first, assuming the mouse doesn't move and the cats travel at equal speed. If the cats arrive at the same time, the mouse will be allowed to move and it will escape while they fight.

You are given 1 query in the form of x, y, and z representing the respective positions for cats A and B, and for mouse C. Write the program to print appropriate answer, which will be printed on a new line.

- If cat A catches the mouse first, print Cat A.
- If cat B catches the mouse first, print Cat B.
- If both cats reach the mouse at the same time, print Mouse C as the two cats fight and mouse escapes.

Input format:

Three space-separated integers describing the respective values of x (cat A's location), y(cat B's location), and z(mouse C's location).

Constraints:

1 <= x,y,z <= 100

Output Format:

return Cat A if cat A catches the mouse first,
Cat B if cat B catches the mouse first, or
Mouse C if the mouse escapes.

Test Case:

Input -1

1 2 3

Output -1

Cat B

Explanation:

The positions of the cats and mouse are shown below:

Cat B will catch the mouse first, so we print Cat B on a new line.

Solution

```
#include<stdio.h>
#include<math.h>
int main()
{
  int a,b,c;
  scanf("%d %d%d",&a,&b,&c);
  if(a>c)
  {
      d=a-c;
  }
  else
  {
      d=c-a;
  }
  if(b>c)
  {
      e=b-c;
  }
  else
  {
      e=c-b;
  }
  if(e==d)
  {
      printf("Mouse C");
  }
  else{
      if(d<e)
      {
          printf("Cat A");
      }
      else{
          printf("Cat B")
      }
  }
  return 0;
}
```

WRITE PROGRAMS

1. Write a C program to find the Maximum and Minimum number of the given two numbers.
2. Write a C program to find the given number is negative or not.
3. Write a C program to find the sum of the last two digits of a given number is odd or even .
4. Write a C program to check whether a number is divisible by 5 and 11 or not.
5. Write a C program to check whether a character is alphabet or not.
6. Given the coordinates of the top-left corner (x1,y1) and bottom-right corner(x2,y2) of a rectangle, write a C program to determine whether it can be a square to?