UNIT - II

* control statements :- → Selection / Branching / Conditional
  Iteration / looping / Repetitive

→ Two types of selection :-

① conditional / Decision control statements.

② Unconditional control statements.

① conditional / Decision :-

(i) 'if'

(ii) 'else / if-else'

(iii) 'if-else-if' (or) 'Ladder if'

(iv) 'switch [Multi way]'

② Unconditional :-

(i) 'break'

(ii) 'continue'

(iii) 'goto'

① conditional / Decision :-

(i) if :-

Syntax :-

```
if (test expression)
{
    statement-block;
}
statement-x;
```

Ex :- Program to check given num is +ve or -ve.

```
# include <stdio.h>
int main( )
{
  int num;
  scanf("%d", &num);
  Printf("Enter a no =");
  if(num>0)
     Printf("%d is +ve", num);
  }
  return 0;
}
```

O/P :-

Enter a no = 9

9 is +ve.

(ii) if-else :-

syntax :-

```
if (test expression)
{
   statement block-1;
}
else
{
   statement block-2;
}
statement -x;
```

Ex :- program for even or odd num.

```
# include <stdio.h>
int main( )
{
   int num;
   printf("Enter a no =");
   scanf ("%d", &num);
   if (num % 2 == 0)
      printf("%d is even", num);
   else
      printf("%d is odd", num)
   return 0;
}
```

o/p :-

Enter a no = 8

8 is even,

Ex :- leap year.

```
#include<stdio.h>
int main( )
{
  int year;
  printf("Enter a year =");
  scanf("%d", &year);
  if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
      Printf("%d is leap year\n", year);
  else
      printf("%d is not leap year", year);
}
```

o/p :-

Enter a year = 1996

1996 is leap year

(iii) if-else ~~if~~ :- [Nested]

syntax :-

```
if (test expression1)
{
  if (test expression2)
  {
    statement block-1;
  }
  else
  {
    statement -2;
  }
}
else
{
   statement -3;
}
statement - x;
```

O/P :-

Enter 3 nums = 9, 5, 17

17 is big.

Ex :- biggest among 3 numbs

```
#include <stdio.h>
int main()
{
  int a, b, c;
  Printf("Enter 3 nums=");
  scanf("%d %d %d", &a, &b, &c
  if(a>b)
  {
    if(a>c)
      printf("%d is big", a);
    else
      printf("%d is big", c);
  }
  else
  {
    if(b>c)
      printf("%d is big", b);
    else
      printf("%d is big", c);
  }
  return 0;
}
```

(iv) if-else-if / Ladder :-

syntax :-

```
if (test expression 1)
{
  statements block -1;
}
else if (test expr 2)
{
  statements block-2;
}
  ---------
  ---------
else if (test expr N)
{
  statement block -N;
}
else
{
  default statements;
}
```

Ex:- student grades using else if

```
#include <stdio.h>
int main()
{
  int tel, hin, eng, mat, sci, soc, total, avg;
  Printf("Enter marks of 6 subjects:");
  scanf("%d %d %d %d %d %d", &tel, &hin, &eng, &mat, &sci, &soc);
  total = tel + hin + eng + mat + sci + soc;
  Printf("Total = %d", total);
  avg = total/6
  Printf("Average = %d\n", avg);
  if (avg > 90)
     Printf("Grade is Excellent");
  else if (avg > 80)
     Printf("A-grade");
  else if (avg > 70)
     Printf("B-grade");
  else if (avg > 60)
     Printf("C-grade");
  else if (avg > 50)
     Printf("D-grade");
  else if (avg > 40)
     Printf("E-grade");
  else if (avg > 30)
     Printf("Fail");
  return 0;
}
```

O/p:-

Enter marks of 6 subjects:
90 90 98 94 95 98
Total marks = 565
Average = 94
Grade is Excellent

(iv) switch statement:-

```
switch (variable)
{
  case constant 1:
     statements;
     break;
  case constant 2:
     statement
     break;
  ------------
  case constant N:
     statement
     break;
  default:
     default statement;
}
```

Ex:-

```
#include <stdio.h>
int main()
{
  int alphabets;
  print("choose 1-26");
  Scanf("%d", &alphabets);
  switch (alphabets)
  {
    case 1:
      printf("A")
      break;
    case 2:
      Printf("B")
      break;
  ------------
  } case N: default statement
```

* Looping control statements / Iterations :

① While loop → Entry control

② Do-while → Exit control

③ For loop

Advantages :

→ Reduce Length of code

→ Less time consuming.

→ Take Less memory space

→ Burden on developer is reducing.

① While Loop :

Syntax :

```
Initialization
while (test expression)
{
    Body of the loop;
    Increment / decrement
}
←statement-x;
```

* Ex : program to print 1 to 10 numbers.

```
#include <stdio.h>
int main()
{
  int i=1;
  while(i<=10)
  {
    printf("%d\n", i);
    i++;
  }
  return 0;
}
```

O/p :

1 2 3 4 5 6 7 8 9 10

* Ex:- Program to print reverse of given number.

```
#include <stdio.h>
int main( )
{
  int i, rem;
  Printf("Enter a number =");
  scanf ("%d", &i);
  Printf ("The reverse number is",);
  while (i!=0)
  {
     rem=i%10;
     Printf ("%d", rem);
     i= i/10;
  }
  Printf("\n");
  return 0;
}
```

O/p:-
Enter a number=14
The reverse no is 41

* Ex:- Program to check given no is Armstrong or not.

A)

```
#include <stdio.h>
int main( )
{
  int n, rem, sum=0, temp;
  Printf(" Enter a number:");
  scanf ("%d", &n);
  temp=n;
  while (n!=0)
  {  rem=n%10;
     sum=sum+(rem*rem*rem);
     n=n/10;
  }
  if (sum==temp)
     printf("%d is Armstrong\n", temp);
  else
     printf ("%d is not Armstrong\n", temp);
  return 0;
}
```

O/p:-
Enter a number is
153 is Armstrong

* Ex :- sum of individual digits.

```
#include <stdio.h>
int main()
{
   int main n, rem, sum=0;
   Printf("Enter a no=\n");
   scanf("%d",&n);
   while(n!=0)
   {
      rem=n%.10;
      sum= sum+rem;
      n=n/10;
   }
   Printf("The sum of individual digits is %d", sum);
   return 0;
}
```

output:-

Enter a no = 126
The sum of individual digits is 9.

② do while :-

→ 'do-while' & 'while' is similar but 'dowhile' runs 'atleast once' even if test condition, is false at first time.

syntax:-

```
do
{
   body of the loop;
}
while (expression);
statement-x;
```

Ex:-

```
#include <stdio.h>
int main()
{
    int num=1;
    do
    {
        printf("%d\n", num);
        num++;
    } while(num<=5);
    return 0;
}
```

output:- 1 2 3 4 5

*

While Loop	Do-while Loop
① condition is checked 'before' entering into loop	① condition is checked 'after' executing in loop
② 'Top-tested' (or) 'entry-controlled loop'	② 'Bottom-tested' (or) 'exit-controlled loop'
③ Minimum iterations are '0'	③ Minimum iterations are '1'
④ Maximum iterations are 'N+1'	④ Maximum iterations are 'N'
⑤ General loop statement	⑤ General loop statement but well suited for menu-driven apps.

*

For loop	Do-while loop
① Loop index can be intialized & altered within loop	① Index will be initialized outside the loop
② 'Very flexible' in Nature	② 'Not so flexible'
③ condition is checked at Beginning of the loop.	③ condition is checked after end of loop
④ Min iterations are '0.'	④ Min iterations are '0' (while) Min iterations are '1' (Do-while)