* Control statements (Decision Control structure)-

i) If statement -

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

An 'if' statement consist of a boolean expression followed by one or more statements.

If the boolean expression evaluates to true then the block of code inside the 'if' statement will be executed.

Eg:-

```
#include <stdio.h>
int main()
{ int a = 10, b = 20;
  if (a < b)
    { printf("a is less than b");
    }
  return 0;
}
```

ii) If ---- else statement -

An 'if' statement can be followed by an optional 'else' statement, which executes when the boolean expression is false.

If the boolean expression evaluates to true, then 'if' block of code will be executed, otherwise 'else' block of code will be executed.

Syntax

```
if (condition)
{
    statement;
}
else
{
    statement;
}
```

Eg:- 1) To check a number is even or odd-

```
#include <stdio.h>
void main()
{
    int n;
    printf("Enter a number");
    scanf("%d", &n);
    if (n%2 == 0)
        printf("Even number");
    else
        printf("Odd number");
}
```

2) Consonent or vowel program-

```
#include <stdio.h>
int main()
{ char ch;
  printf("Enter any alphabet\n");
  scanf("%c", &ch);
  if (ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')
    printf("Vowel");
  else
    printf("Consonent");
```

iii) Else if ladder (if --- else if --- else) :-

An 'if' statement can be followed by an optional 'else if --- else' statement, which is very useful to test various conditions.

In this type, 'if' condition is false control pass to block where condition is again checked with its 'if statement'.

Syntax

```
if (condition)
  {
    statements;
  }
else if (condition)
  {
    statement 2;
  }
else if (condition)
  {
    statement 3;
  }
else
  {
    statement 4;
  }
```

Eg:- Program to check given no. is +ve, -ve or zero

```
# include <stdio.h>
void main()
{ int a;
printf("Enter any number\n");
scanf("%d", &a);
if (a<0)
printf("Negative");
```

Note - In if block, ~~that~~ only one statement needed then not necessary to give curly ~~braces~~

```
else if (a > 0)
printf ("Positive");
else
   printf ("zero");
```

ii)
```
# include <stdio.h>
void main()
{ int x = 10;
  if (x > 5)
  printf ("x is greater than 5");
  else if (x < 8)
  printf ("x is less than 8");
  else if (x == 10)
  printf ("x is equal to 10");
  else
   printf ("No one condition is true");
}
```

iv) Nesting of if ... else -

When there are another if else stement in if - block, then it is called nesting of if - else ~~stement~~ statement

syntax

```
if (condition)
  {
    if (condition)
      statement1;
    else
      statement2;
  }
  else
  {
    if (condition)
    statement3;
    else statement4;
  }
```

Eg:- 1) Greater value in 2 numbers -

```
# include <stdio.h>
int main()
{
  int a, b;
 printf("Enter first number\n");
 scanf("%d", &a);
printf("Enter second number\n");
 scanf("%d", &b);
 if (a > b)
   printf("first no is greater");
 if (a < b)
    printf("Second no. is greater");
 if (a == b)
    printf("Both are equal");
}
```

2) Greater value in 3 numbers -

```
# include <stdio.h>
~~in~~ void main()
{  int a, b, c;
 printf("Enter the first no.\n");
   scanf("%d", &a);
 printf("Enter the second no.\n");
   scanf("%d", &b);
 printf("Enter the third no.\n");
   scanf("%d", &c);
 if (a>b && a>c)
    printf("%d is greater", a);
 if (b>a && b>c)
    printf("%d is greater", b);
 if (c>a && c>b)
    printf("%d is greater", c);
}
```

Date
Page

ii) Switch statement -

Switch statement allows us to execute one statement from many statement and that statements are called case. Actually in switch statements, inside the body of switch a number of cases are used and a parameter are passed and from which case this parameter is matched, executed.

Syntax

Parameter

Keyword

```
switch (*)
{
  case 1:
    statement 1;
    break;
  case 2:
    statement 2;
    break;
  ---------
  ---------
  case n:
    statement n;
    break;
  default:
    default statement;
}
```

Note -

- In the switch statement a value/number is passed in the place of parameter and that case will execute which is equal to that value/number.

- If no case matched with parameter then default case will execute.

Eg:- Input any number and print day of week.

```
# include <stdio.h>
void main()
{ int n;
printf ("Enter any number\n");
scanf ("%d", &n);
switch (n);
 {
    case 1:
   printf ("Monday");
     break;
   Case 2:
   printf ("Tuesday");
     break;
   Case 3:
     printf ("Wednesday");
       break;
   Case 4:
     printf ("Thursday");
        break;
   Case 5:
     printf ("Friday");
        break;
   Case 6:
     printf ("Saturday");
        break;
   Case 7:
     printf ("Sunday");
        break;
   default:
      printf ("Invalid input");
 }
}
```

Date
Page

* Un-conditional transfer statement-

i) Goto statement-

'Goto' is an un-conditional control transfer statement, whenever encounter this statement, the control jumps to said line-lable in the program.

Syntax

```
lable:
goto line-lable; // line label may be no. or characters
```

* Table of any number using goto statement-

```
#include <stdio.h>
int main()
{
  int num, i=1;
  printf("Enter any number");
  scanf("%d", &num);
  table:
  printf("%d x %d = %d\n", num, i, num*i);
    i++;
  if (i<=10)
  goto table;
}
```

* Input five no. and print sum using goto statement-

```
#include <stdio.h>
int main()
{ int count=0, n, s=0;
  start:
    print("Enter a no.");
      scanf("%d", &n);
      s = s+n, count++;
  if (count<5)
    goto start;
printf("\n sum of numbers = %d", sum);
```

ii) Break statement -

This statement is used either in switch block or loop. Whenever encounter this statement the control exit from the block in which it has defined.

Syn- Break;

Eg:-
```
# include <stdio.h>
int main()
{ int j;
  for(j=0; j<6; j++)
  {if (j==4)
    break;
  printf("%d", j);}
```

Output - 0, 1, 2, 3

iii) Continue statement -

It is used for continuing next iteration of loop after skipping some statement of loop. When it encountered control automatically passes through the beginning of the loop.

Eg:-
```
# include <stdio.h>
int main()
{ int n;
  for (n=2; n<=9; n++)
  { if(n==4)
    continue;
  printf("%d", n);
  }
```

Output- 2, 3, 5, 6, 7, 8, 9

Date
Page

* Loops in C (Looping statement) -

Loop -

It is a block of statement that performs set of instructions. In loops, repeating particular portion of the program either a specified number of time or untill a particular no. of condition is being satisfied.

"The looping is a process of repeating a single statement or a group of statements untill some condition for termination of the loop is satisfied"

* Types of loops -

i) Entry loop -

Those loop in which condition is checked before the execution of the statement. Thus if the condition is false in the beginning the loop will not run even once.

Eg:- for loop, while loop.

ii) Exit loop -

Those loops in which the condition is checked after the execution of the statement. Thus if the condition is false in the beginning the loop will run at least once.

Eg:- do while loop

* There are three loops in C -

i) While loop -

When we want to do something a fixed no. of time but not known about the number of iteration, in a program then while loop is used.

In this loop, first condition is checked if, it is true, body of the loop is executed otherwise control will be come out of loop.

syntax -

```
while (condition)
{
    statements;
}
```

* Table of one -

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

ii) Print 'Hello' 5 times -

```
#include <stdio.h>
int main()
{
  int a = 1;
  while (a <= 5)
  {
    printf("Hello\n");
      a = a+1;
  }
}
```

iii) Do-while loop -

It is also used for looping and it is a type of exit loop.

syntax

```
do
{
  statement;
}
while (condition);
```

Eg:- Table of one -

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

iii) For loop -

This loop is generally used when number of iteration are known in advance.

syntax

```
for(initialization ; condition ; increment/dec)
{
    statements;
}
```

Eg:- Table of one -

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

Table of any number -

```
#include <stdio.h>
int main()
{ int n;
  printf("Enter any num\n");
    scanf("%d", &n);
  for (int i=1; i<=10; i++)
    {
      printf("%d", i*n);
    }
}
```