C – LANGUAGE

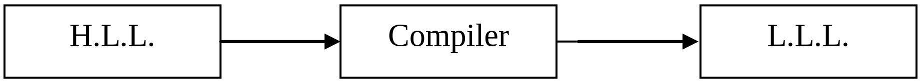
INTRODUCTION

In 1972 Denice Richie developed C Language in BELL Laboratories (USA).

<u>Def: -</u> Middle level language.

<u>Use: -</u> General purpose.

<u>Compiler: -</u> It converts H.L.L. to L.L.L



English
Ex:- C, C++,java,
Pascal, Basic, Fortran….. etc.

Binary
Ex :- 0’s & 1’s

<u>Key words: -</u> These are the words pre-define by the programmers.

int
float
char
short
long
double
signed
unsigned

if
else
while
do
for
switch
break
case

continue
default
void
return
struct
union
enum
sizeof

auto
register
static
external
typedef

<u>Character set: -</u>

A – Z
a – z
0 – 9

Special characters: -
$! & ^ ¦ * / % + - { } ( ) [ ] : ; . , ’ ” \ = > < ? ……. etc

<u>Operators: -</u>

1. Arithmetical operators : + - * / %
2. Logical operators : &&, ¦¦
3. Relational operators : < > <= >= = = !=
4. Conditional operator : ?:
5. Bit wise operators : & ¦ ^ << >>
6. Unary operators : ++, --
7. Assignment operators : +=, -=, *=, /=, %=
8. Specail operators : , . ’ ” ;

Data Types: -

1. Primary data types - int, char, float
2. User defined data types - struct, union, enum
3. Derived data types - pointer, array
4. Empty data types - void

1. Simple Program.

```
#include<stdio.h>
void main( )
{
        printf("welcome");
}
```

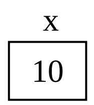
Output:

```
welcome
```

Alt + F9 → compile
Ctrl + F9 → run
Alt + F5 → see the output

2. Simple Program about printf

```
#include<stdio.h>
void main( )
{
        int x=10;
        clrscr( );
        printf("x :%d",x);
        getch( );
}
```

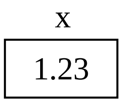


Output:

```
x : 10
```

3. Simple Program about printf

```
#include<stdio.h>
void main( )
{
        float x=1.23;
        clrscr( );
        printf("x :%f",x);
        getch( );
}
```

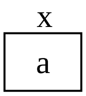


Output:

```
x : 1.230000
```

4. Simple Program about printf

```
#include<stdio.h>
void main( )
{
        char x='a';
        clrscr( );
        printf("x :%c",x);
        getch( );
}
```

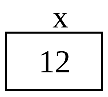


Output:

```
x : a
```

5. Simple Program about scanf

```
#include<stdio.h>
void main( )
{
        int x;
        clrscr( );
        printf(“enter no:”);
        scanf(“%d”,&x);
        printf(“x :%d”,x);
        getch( );
}
```

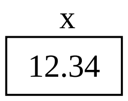


Output:

```
enter no: 12
x :12
```

6. Simple Program about scanf

```
#include<stdio.h>
void main( )
{
        float x;
        clrscr( );
        printf(“enter no:”);
        scanf(“%f”,&x);
        printf(“x :%f”,x);
        getch( );
}
```

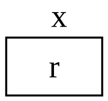


Output:

```
enter no:12.34
x :12.340000
```

7. Simple Program about scanf

```
#include<stdio.h>
void main( )
{
        char x;
        clrscr( );
        printf(“enter char :”);
        scanf(“%c”,&x);
        printf(“x :%c”,x);
        getch( );
}
```

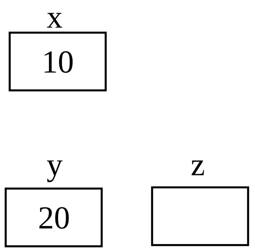


Output:

```
enter char : r
x :r
```

8. Write a program to add two nos.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int  x,y,z;
        clrscr( );
        printf(“enter two nos:”);
        scanf(“%d%d”,&x,&y);
        z = x + y;
        printf(“z :%d”,z);
        getch( );
}
```

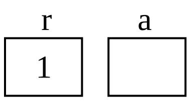


Output:

```
enter two nos:10  20
z :30
```

9. Write a program to print area of circle. ($a=\Pi r^2$)

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int  r;
        float a;
        clrscr( );
        printf(“enter r :”);
        scanf(“%d”,&r);
        a = 3.14 * r * r;
        printf(“a :%f”,a);
        getch( );
}
```

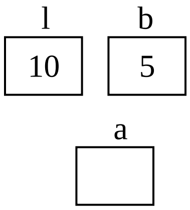


Output:

```
enter r :1
a : 3.140000
```

10. Write a program to print area of rectangle. (a=lb)

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int  l,b,a;
        clrscr( );
        printf(“enter l b :”);
        scanf(“%d%d”,&l,&b);
        a = l * b;
        printf(“a :%d”,a);
        getch( );
}
```

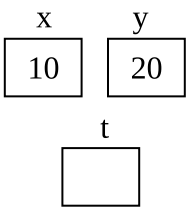


Output:

```
enter l b : 10     5
a : 50
```

11. Write a program swapping or interchange two numbers. (using third variable)

```
void  main( )
{
        int  x,y,t;
        clrscr( );
        printf(“enter two nos :”);
        scanf(“%d%d”,&x,&y);
        printf(“x :%d  y :%d \n”,x,y);

        t = x;
        x = y;
        y = t;

        printf(“x :%d  y :%d \n”,x,y);
        getch( );
}
```

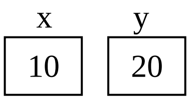


Output:

```
enter two nos: 10  20
x :10    y : 20
x :20    y : 10
```

11. Write a program swapping or interchange two numbers. (without using third variable)

```
void  main( )
{
        int  x,y;
        clrscr( );
        printf(“enter two nos :”);
        scanf(“%d%d”,&x,&y);
        printf(“x :%d  y :%d \n”,x,y);
        x = x + y;
        y = x - y;
        x = x - y;
        printf(“x :%d  y :%d \n”,x,y);
        getch( );
}
```

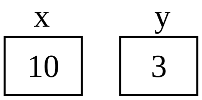


Output:

```
enter two nos: 10    20
x : 10    y : 20
x : 20   y  : 10
```

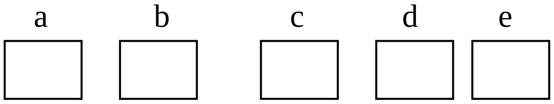
12. Write a program to add, sub, multi, divide & find remainder of the two nos.

```
void main( )
{
        int  x,y,a,b,c,e;
        float d;
        clrscr( );
        printf(“enter two nos:”);
        scanf(“%d%d”,&x,&y);
        a = x + y;
        b = x – y;
        c = x * y;
        d = x / (float) y;
        e = x % y;
        printf(“a :%d   b : %d  c : %d \n”,a,b,c);
        printf(“d :%f   e  : %d ”,d,e);
        getch( );
}
```



Output:

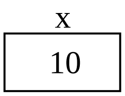
```
 enter two nos: 10   3
a : 13    b : 7    c : 30
d : 3.333333    e : 1
```



Increment & Decrement:

13. Pre-increment & Post-increment

```
void   main( )
{
        int  x = 10;
        clrscr( );
        printf(“x :%d\n”,x);
        printf(“x :%d\n”,x++);
        printf(“x :%d\n”,x);
        printf(“x :%d\n”,++x);
        printf(“x :%d\n”,x);
        getch( );
}
```



Output:

```
x :10
x :10
x :11
x :12
x :12
```

14. Pre-increment & Post-increment

```
void   main( )
{
        int  x = 10;
        clrscr( );
        printf("x :%d\n",x);
        printf("x :%d\n",++x);
        printf("x :%d\n",x);
        printf("x :%d\n",x++);
        printf("x :%d\n",x);
        getch( );
}
```

x

$\boxed{10}$

Output:

```
x :10
x :11
x :11
x :11
x :12
```

15. Pre-decrement & Post-decrement

```
void   main( )
{
        int  x = 10;
        clrscr( );
        printf("x :%d\n",x);
        printf("x :%d\n",x--);
        printf("x :%d\n",x);
        printf("x :%d\n",--x);
        printf("x :%d\n",x);
        getch( );
}
```

x

$\boxed{10}$

Output:

```
x :10
x :10
x :9
x :8
x :8
```

16. Pre-decrement & Post-decrement

```
void   main( )
{
        int  x = 10;
        clrscr( );
        printf("x :%d\n",x);
        printf("x :%d\n",--x);
        printf("x :%d\n",x);
        printf("x :%d\n",x--);
        printf("x :%d\n",x);
        getch( );
}
```

x

$\boxed{10}$

Output:

```
x :10
x :9
x :9
x :9
x :8
```

Conditional Operator: ? :

1.
2. (exp) ? Statement1 : Statement2;
3. varialble = (exp) ? value1 : value 2;

Write a program check the given number is even or odd.

```
void main( )                                        Output:
{
        int x;                                      enter no: 12
        clrscr( );                                  12 is  +ve
        printf(“enter no:”);
        scanf(“%d”,&x);
        (x>0)? printf(“%d is +ve”,x) : printf(“%d is -ve”,x);
        getch( );
}
```

Write a program check the given number is even, odd or zero.

```
void main( )                                        Output:
{
        int x;                                      enter no: -12
        clrscr( );                                  -12 is  -ve
        printf(“enter no:”);
        scanf(“%d”,&x);
        (x>0)? printf(“%d is +ve”,x) : (x<0) ? printf(“%d is -ve”,x) : printf(“0”);
        getch( );
}
```

Write a program to printf y=1 when x > 0, y=-11 when x < 0.

```
void main( )                                        Output:
{
        int x,y;                                    enter no: 12
        clrscr( );                                  x:12  y:1
        printf(“enter no:”);
        scanf(“%d”,&x);
        y = (x>0)? 1 : 0;
        printf(“x :%d y :%d”,x,y);
        getch( );
}
```

Write a program to evaluate expressions y1=logx, y2=e^x, y3=√x, y4=|x|, y5=x^n

```
#include<math.h>
void main( )                                   Output:
{
        int x,n,y4,y5;                         enter x n values :2 5
        float y1,y2,y3;                        y1:    y2:   y3:1.414000 y4: 2 y5:32
        clrscr( );
        printf(“enter x n values :”);
        scanf(“%d%d”,&x,&n);
        y1=log(x);              y2=exp(x);
        y3=sqrt(x);             y4=abs(x);      y5=pow(x,n);
        printf(“y1:%f  y2:%f  y3:%f  y4:%d  y5:%d”,y1,y2,y3,y4,y5);
        getch();
}
```

Write a program to evaluate expressions y1=sinx, y2=cosx, y3=tanx

```
#include<math.h>
void main( )
{
        int x;
        float y1,y2,y3;
        clrscr( );
        printf(“enter x value :”);
        scanf(“%d”,&x);
        y1=sin(x*3.14/180);
        y2= cos(x*3.14/180);
        y3= tan(x*3.14/180);
        printf(“y1:%f  y2:%f  y3:%f”,y1,y2,y3);
        getch();
}
```

Output:

```
enter x:30
y1:     y2:     y3:
```

LOOPS

Def: -

One or more that one statement(s) are repeated or executed through a condition.

Types of loops: -

I. Conditional statements
   1. if
   2. switch

II. Iterative statements
   3. while
   4. do-while
   5. for

III. Un-conditional statement
   1. goto

If: -

Def: -

Check the condition, when condition is true execute if following statement(s).

Syntax: -

```
        if(cond)                              if(cond)
                statement1;    (or)           {
                                                 ┌──────┐
                                                 │ B.S. │
                                                 └──────┘
                                              }
```

1. Write a program to check the given no is positive. (using if)

```
void   main( )
{                                                         Output:
        int  x;                        x
        clrscr( );                  ┌──────┐              enter any no : 10
        printf("enter any no :");   │  10  │              10 is +ve
        scanf("%d",&x);             └──────┘
        if(x>0)
                printf("%d is +ve",x);
        getch( );
}
```

2. Write a program to check the given no is even. (using if)

```
void   main( )
{
        int  x;                        x                  Output:
        clrscr( );                  ┌──────┐
        printf("enter any no :");   │  20  │              enter any no : 20
        scanf("%d",&x);             └──────┘              20 is even
        if(x% 2 = = 0)
                printf("%d is even ",x);
        getch( );
}
```

If-else: -

Def: -

Check the condition, when condition is true execute if following statement(s). When condition is false else following statement(s) will be executed.

Syntax: -

```
        if(cond)                                if(cond)
                statement1;     (or)            {
        else                                        B.S.
                statement2;                     }
                                                else
                                                {
                                                    B.S.
                                                }
```

1. Write a program to check the given no is positive or negative. (using if-else)

```
void   main( )
{
        int  x;                         x                   Output:

        clrscr( );                    12
        printf(“enter any no :”);                           enter any no : 12
        scanf(“%d”,&x);                                     12 is +ve
        if(x>0)
                printf(“%d is +ve”,x);                      enter any no : -2
        else                                                -2 is -ve
                printf(“%d is -ve”,x);
        getch( );
}
```

2. Write a program to check the given no is even or odd. (using if-else)

```
void   main( )
{
        int  x;                         x                   Output:
        clrscr( );                     10
        printf(“enter any no :”);                           enter any no : 10
        scanf(“%d”,&x);                                     10 is even
        if(x% 2 = = 0)
                printf(“%d is even ”,x);                    enter any no : 15
        else                                                15 is odd
                printf(“%d is odd ”,x);
        getch( );
}
```

3. Write a program to print greatest of two nos. (using if-else)

```
void main( )
{
        int  x,y;
        clrscr( );
        printf("enter two nos:");
        scanf("%d%d",&x,&y);
        if ( x > y)
                printf("%d", x);
        else
                printf("%d", y);
        getch( );
}
```

x	y
20	10

Output:

```
enter two nos: 20 10
20

enter two nos: 10 50
50
```

4. Write a program to check the given two nos are equal or not equal. (using if-else)

```
void main( )
{
        int  x,y;
        clrscr( );
        printf("enter two nos:");
        scanf("%d%d",&x,&y);
        if ( x = = y)
                printf(" equal ");
        else
                printf("not equal");
        getch( );
}
```

x	y
10	10

Output:

```
enter two nos: 10   10
equal

enter two nos: 10   5
not equal
```

Else-if lader: -

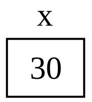
If-else is following with another if-else.

Syntax: -

```
        if(cond)
                statement1;
        else
        if(cond)
                 statement2;
        else
        if(cond)
                 statement3;
        else
        if(cond)
                 :
                :
                :
```

Write a program to chech the given number is +ve, -ve or zero (0).

```
 void main( )
{
        int  x;
        clrscr( );
        printf(“enter  no:”);
        scanf(“%d”,&x);
        if ( x>0)
                printf(“%d is +ve”, x);
        else
        if(x<0)
                printf(“%d is -ve”, x);
        else
                printf(“0”);
        getch( );
}
```

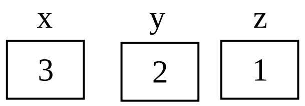


Output:

```
enter no: 30
30 is +ve
```

Write a program to print greatest of three nos.

```
void main( )
{
        int  x,y,z;
        clrscr( );
        printf(“enter three nos:”);
        scanf(“%d%d%d”,&x,&y,&z);
        if ( x > y &&  x > z )
                printf(“%d”, x);
        else
        if(y > z)
                printf(“%d”, y);
        else
                printf(“%d”,z);
        getch( );
}
```



Output:

```
enter three nos: 3   2   1
3

enter three nos: 2   3   1
3
```

Nested-if: -

If contains anether if or if follows anether if.

Syntax: -

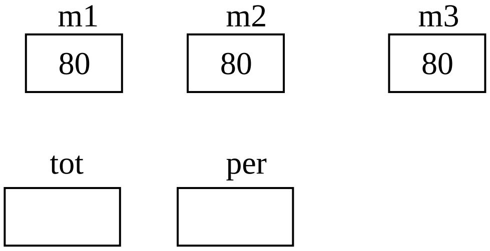
```
        if(cond)
        {
                if (cond)
                        st1;
        }
```

Write a program to find total, average & grade of the student when three subjects are given.

```
void main( )
{
        int  m1,m2,m3,tot;
        float per;
        clrscr( );
        printf(“enter three sub marks :”);
        scanf(“%d%d%d”,&m1,&m2,&m3);
        tot = m1 + m2 + m3;
        per = tot / 3.0;
        printf(“ m1 : %d\n”,m1);
        printf(“ m2 : %d\n”,m2);
        printf(“ m3 : %d\n”,m3);
        printf(“ tot : %d\n”,tot);
        printf(“ per : %f\n”,per);
        if( m1 >= 35 && m2 >= 35 && m3 >= 35 )
        {
                if( per >= 75)
                        printf(“ distinction “);
                else
                if( per >= 60)
                        printf(“ 1st class “);
                else
                if( per >= 50)
                        printf(“ 2nd class “);
                else
                        printf(“ 3rd class “);
        }
        else
                printf(“ fail “);
        getch( );
}
```

Output:

```
enter three sub marks: 80  80  80
m1  : 80
m2  : 80
m3  : 80
tot   : 240
per  : 80.000000
distinction
```



While: -

Def: -

Check the condition, when condition is true execute the loop until the condition is false.

Syntax: -

```
while(cond)                        while(cond)
        st1;            (or)       {
                                     B.S.
                                   }
```

1. Write a program to print 1,2,3,……… 10

```
void   main( )
{                                                          Output:
        int  i=1;                              i
        clrscr( );                             1           1
        while(i<=10)                                       2
        {                                                  3
                printf("%d\n",i);                          :
                i++;                                       :
        }                                                  10
        getch( );
}
```

2. Write a program to print 2,4,6,8,10

```
void   main( )
{                                           i              Output:
        int  i=2;                           2
        clrscr( );                                         2
        while(i<=10)                                       4
        {                                                  6
                printf("%d\n",i);                          8
                i=i+2;                                     10
        }
        getch( );
}
```

3. Write a program to print 1,3,5,7,9

```
void   main( )                                             Output:
{                                          i
        int  i=1;                          1               1
        clrscr( );                                         3
        while(i<=10)                                       5
        {                                                  7
                printf("%d\n",i);                          9
                i=i+2;
        }
        getch( );
}
```

4. Write a program to print 10, 9, 8, …….. 1

```
void   main( )                  i                    Output:
{                              10
        int  i=10;                                   10
        clrscr( );                                   9
        while(i>=1)                                  8
        {                                            :
                printf("%d\n",i);                    :
                i--;                                 1
        }
        getch( );
}
```

5. Write a program to print 1+2+3+……… +10

```
void   main( )
{
        int  i=1,sum=0;
        clrscr( );
        while(i<=10)
        {
                sum = sum + i;
                i++;
        }
        printf(“%d”,sum);
        getch( );
}
```

i	sum
1	0

Output:

```
55
```

6. Write a program to print 1,2,3,……… n

```
void   main( )
{
        int  i=1,n;
        clrscr( );
        printf(“enter n :”);  scanf(“%d”,&n);
        while(i<=n)
        {
                printf(“%d\n”,i);
                i++;
        }
        getch( );
}
```

i
1

Output:

```
1
2
3
:
:
:
10
```

7. Write a program to print reverse of the given number.

```
void   main( )
{
        int  num, rem, sum=0;
        clrscr( );
        printf(“enter the number :”);
        scanf(“%d”,&num);
        while(num>0)
        {       rem = num % 10;
                sum = (sum * 10)+ rem;
                num = num / 10;
        }
        printf(“reverse o the given number is :%d”,sum);
        getch( );
}
```

num	rem	sum
123		0

Output:

```
enter the number : 123
reverse o the given number is : 321
```

8. Write a program to check the given number is palindrome or not.

```
void   main( )
{
        int  num, rem, sum=0,temp;
        clrscr( );
        printf(“enter the number :”);   scanf(“%d”,&num);
```

num	rem	sum	temp
121		0	121

```
        temp=num;
        while(num>0)
        {       rem = num % 10;
                sum = (sum * 10)+ rem;
                num = num / 10;
        }
        if(sum = = temp)
                printf("%d is palindrome ",temp);
        else
                printf("%d is not palindrome ",temp);
        getch( );
}
```

Output:

```
enter the number : 121
121 is palindrome
```

9. Write a program to check the given number is Armstrong or not.

```
void   main( )
{
        int  num, rem, sum=0,temp;
        clrscr( );
        printf("enter the number :");
        scanf("%d",&num);
        temp=num;
        while(num>0)
        {       rem = num % 10;
                sum = sum + (rem*  rem * rem);
                num = num / 10;
        }
        if(sum = = temp)
                printf("%d is Armstrong ",temp);
        else
                printf("%d is not Armstrong ",temp);
        getch( );
}
```

num	rem	sum	temp
153		0	153

Output:

```
enter the number : 153
153 is Armstrong
```

10. Write a program to print sum of the digits of the given numbers.

```
void   main( )
{
        int  num, rem, sum=0;
        clrscr( );
        printf("enter the number :");
        scanf("%d",&num);
        while(num>0)
        {
                rem = num % 10;
                sum = sum + rem;
                num = num / 10;
        }
        printf("%d",sum);
        getch( );
}
```

num	rem	sum
123		0

Output:

```
enter the number: 123
6
```

For: -

Syntax: -

```
for(Exp1;Exp2;Exp3)        (or)        for(Exp1;Exp2;Exp3)
        statement1;                    {

                                           B.S.

                                       }
```

Exp1: - (Initialization)

Ex: -

i=0; j=10; k=1; …….. etc.

Exp2: - (Condition)

Ex: -

i<=10; j>=1; k<=100; …….. etc.

Exp3: - (Increment / Decrement)

Ex: -

i++; j--; k++; …….. etc.

1. Write a program to print 1, 2, 3, ……… 10

```
void main( )                                   Output:
{                                    i
        int i;                       1           1
        clrscr( );                               2
        for(i=1;i<=10;i++)                       3
                printf("%d \n",i);               :
        getch( );                               10
}
```

2. Write a program to print 2, 4, 6, ……… 10

```
void main( )                           i               Output:
{                                      2               2
        int i;                                         4
        clrscr( );                                     6
        for(i=2;i<=10;i=i+2)                           8
                printf("%d \n",i);                     10
        getch( );
}
```

3. Write a program to print 1, 3, 5, ……… 9

```
void main( )                                           Output:
{                                      i               1
        int i;                         1               3
        clrscr( );                                     5
        for(i=1;i<=10;i=i+2)                           7
                printf("%d \n",i);                     9
        getch( );
}
```

4. Write a program to print 1, 4, 9, ……… 100

```
void main( )
{
        int i;
        clrscr( );
        for(i=1;i<=10;i++)
                printf(“%d \n”,i*i);
        getch( );
}
```

i $\boxed{1}$

Output:

```
1
4
9
:
100
```

5. Write a program to print 10, 9, 8, ……… 1

```
void main( )
{
        int i;
        clrscr( );
        for(i=10;i>=1;i--)
                printf(“%d \n”,i);
        getch( );
}
```

i $\boxed{10}$

Output:

```
10
9
8
:
1
```

6. Write a program to print 1+ 2 + 3+ ………+10

```
void main( )
{
        int i,sum=0;
        clrscr( );
        for(i=1;i<=10;i++)
                sum = sum + i;
        printf(“%d \n”,sum);
        getch( );
}
```

i $\boxed{1}$ sum $\boxed{0}$

Output:

```
55
```

7. Write a program to print 1, 2, 3, ……… n

```
void main( )
{
        int i,n;
        clrscr( );
        printf(“enter n :”);
        scanf(“%d”,&n);
        for(i=1;i<=n;i++)
                printf(“%d \n”,i);
        getch( );
}
```

i $\boxed{1}$ n $\boxed{5}$

Output:

```
enter n : 5
1
2
3
4
5
```

8. Write a program to print 1+ 2 + 3+ ………+n

```
void main( )
{
        int  i,sum=0,n;
        clrscr( );
        printf(“enter n :”);
        scanf(“%d”,&n);
        for(i=1;i<=n;i++)
                sum = sum + i;
        printf(“%d \n”,sum);
        getch( );
}
```

i $\boxed{1}$ sum $\boxed{0}$

Output:

```
enter n : 10
55
```

9. Write a program to print mathematical table.

```
void main( )
{
        int i,n;
        clrscr( );
        printf(“enter which table you want :”);
        scanf(“%d”,&n);
        for(i=1;i<=10;i++)
                printf(“%2d * %2d = %2d \n”,n,i,n*i);
        getch( );
}
```

i	n
1	5

Output:

```
enter which table you want: 5
 5 *  1 =   5
 5 *  2 = 10
 5 *  3 = 15
 :
 :
5 * 10 = 50
```

10. Write a program to print

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
void main( )
{
        int line,i,n;
        clrscr( );
        printf(“enter how many line you want :”);
        scanf(“%d”,&n);
        for(line=1;line<=n;line++)
        {
                for(i=1;i<=line;i++)
                        printf(“%d \t”,i);
                printf(“\n”);
        }
        getch( );
}
```

line
1

i	n
1	5

Output:

```
enter how many line you want : 5
1
2    2
3    3     3
4    4     4     4
5    5     5     5      5
```

11. Write a program to print

```
     1
    2 2
   3 3 3
  4 4 4 4
 5 5 5 5 5
```

```
void main( )
{
        int line,i,j,n;
        clrscr( );
        printf(“enter how many line you want :”);
        scanf(“%d”,&n);
        for(line=1;line<=n;line++)
        {
                for(i=1;i<=n-line;i++)
                        printf(“  ”);
                for(j=1;j<=line;j++ )
                        printf(“%2d”,line);
                printf(“\n”);
        }
        getch( );
}
```

line	n
1	5

i	j
1	1

Output:

```
enter how many line you want : 5

        1
      2   2
    3   3    3
  4   4    4   4
 5    5    5    5   5
```

do-while: -

Def: - Execute the loop, after execution of loop check the condition when condition is true execute the loop until the condition is false.

syntax: -

```
    do
    {
       B.S.
    }while(cond);
```

Difference between while and do-while: -

While	do-while
Def: - Check the condition, when condition is true execute the loop until the condition is false.	Def: - Execute the loop, after execution of loop check the condition when condition is true execute the loop until the condition is false.
Syntax: - while(cond) { B.S. }	**syntax: -** do { B.S }while(cond);
When the condition is true then only execute the loop.	When the condition is true or false loop will be executed at least one's.

1. Write a program to print 1, 2, 3, ….. 10 (using do-while)

```
void   main( )
{                                           i
        int  i=1;                         [ 1 ]                  Output:
        clrscr( );
        do                                                       1
        {                                                        2
                printf("%d\n",i);                                3
                i++;                                             :
        } while(i<=10);                                          :
        getch( );                                                10
}
```

2. Write a program to print sum of the given n numbers.

```
void   main( )
{                                                                Output:
        int  i=1,x,n,sum=0;              i      sum
        clrscr( );                     [ 1 ]   [ 0 ]             enter n : 5
        printf("enter n :");                                     enter any no : 3
        scanf("%d",&n);                   n       x              enter any no : 1
        do                              [ 5 ]   [ 3 ]            enter any no : 4
        {                                                        enter any no : 7
                printf("enter any no :");                        enter any no : 2
                scanf("%d",&x);                                  sum : 17
                sum = sum + x;
                i++;
        } while(i<=n);
        printf("sum :%d",sum);       getch( );
}
```

switch: -

Select one statement from number of statements.

Syntax: -

```
switch (op)
{
                 ‘char’
        case      (or)   :      statement 1;   break;
                 const
                 ‘char’
        case      (or)   :      statement 2;   break;
                 const
                 ‘char’
        case      (or)   :      statement 3;   break;
                 const
                  :
        default :        statement n;
}
```

1. Simple program for switch.

```
void   main( )
{         int  n;                              n
          clrscr( );                          3
          printf(“enter no :”);
          scanf(“%d”,&n);
          switch(n)
          {       case 1 : printf(“ one ”);        break;
                  case 2 : printf(“ two ”);        break;
                  case 3 : printf(“ three ”);      break;
                  case 4 : printf(“ four ”);       break;
                  default : printf(“ aaaa ”);
          }
          getch( );
}
```

Output:

```
enter no : 3
 three
```

2. Simple program for switch.

```
void   main( )
{
          int  n;
          clrscr( );                                n
          printf(“enter no :”);                    3
          scanf(“%d”,&n);
          switch(n)
          {
                  case 1 : printf(“ one ”);
                  case 2 : printf(“ two ”);
                  case 3 : printf(“ three ”);
                  case 4 : printf(“ four ”);
                  default : printf(“ aaaa ”);
          }
          getch( );
}
```

Output:

```
enter no : 3
three  four   aaaa
```

3. Simple program for switch.

```
void   main( )
{       char  x;                          x
        clrscr( );                        b
        printf(“enter  character :”);
        scanf(“%c”,&x);
        switch(x)
        {       case ‘a’ : printf(“ a ”);           break;
                case ‘b’ : printf(“ b ”);           break;
                case ‘c’ : printf(“ c ”);           break;
                case ‘d’ : printf(“ d ”);           break;
                default : printf(“ default statement ”);
        }   getch( );
}
```

Output:

```
enter  character : b
b
```

4. Simple program for switch.

```
void   main( )                    x
{                                 b
        char  x;
        clrscr( );
        printf(“enter  character :”);
        scanf(“%c”,&x);
        switch(x)
        {       case ‘a’ : printf(“ a ”);
                case ‘b’ : printf(“ b ”);
                case ‘c’ : printf(“ c ”);
                case ‘d’ : printf(“ d ”);
                default : printf(“ default statement ”);
        } getch( );
}
```

Output:

```
enter  character : b
b c d default statement
```

5. Write a program for add, subtract, multiple, divide and find remainder of the two numbers when operator is given.

```
void   main( )
{       char  op;
        int x,y,a,b,c,e;     float d;
        clrscr( );
        printf(“enter  character [+ - * / %]:”);          scanf(“%c”,&op);
        printf(“enter two nos :”);                        scanf(“%d%d”,&x,&y);
        switch(op)
        {       case ‘+’ :  a = x+y;            printf(“ %d ”,a);          break;
                case ‘-’ :   b = x-y;           printf(“ %d ”,b);          break;
                case ‘*’ :  c = x*y;            printf(“ %d ”,c);          break;
                case ‘/’ :   d = x/(float) y;   printf(“ %f ”,d);          break;
                case ‘%’ : e= x%y;              printf(“ %d ”,e);          break;
                default : printf(“ default statement ”);
        }       getch( );
}
```

OUTPUT:

```
enter  character [+ - * / %]: *
enter two nos : 10 3
30
```

ARRAYS

Def: -

Collection of similar or related data elements.

Declaration: -

i). Data type Variable[size]={values};

int x[5]={10,20,30,40,50};

0	1	2	3	4
10	20	30	40	50

float a[3] = {1.1, 1.2, 1.3};

0	1	2
1.1	1.2	1.3

char s[10]={‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘i’, ‘j’};

0	1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	h	i	j

ii). Data type Variable[size];

int x[3];

0	1	2

float a[5];

0	1	2	3	4

char b[5];

0	1	2	3	4

1. Simple program about array.

```
void main( )
{
        int x[5]={10,20,30,40,50};
        clrscr( );
        printf(“ x[0] : %d \n”,x[0]);
        printf(“ x[1] : %d \n”,x[1]);
        printf(“ x[2] : %d \n”,x[2]);
        printf(“ x[3] : %d \n”,x[3]);
        printf(“ x[4] : %d \n”,x[4]);
        getch( );
}
```

0	1	2	3	4
10	20	30	40	50

OUTPUT:

```
x[0] : 10
x[1] : 20
x[2] : 30
x[3] : 40
x[4] : 50
```

2. Write a program read five numbers store in array and print those five numbers.

```
void main( )
{
        int x[5],i;
        clrscr( );
        printf(“ enter 5 eles : ”);
        for(i=0;i<=4;i++)
                scanf(“%d”,&x[i]);
        printf(“eles :”);
        for(i=0;i<=4;i++)
                printf(“ %d \t”,x[i]);
        getch( );
}
```

0	1	2	3	4
10	20	30	40	50

OUTPUT:

```
enter 5 eles : 10  20  30  40  50
eles : 10     20    30   40    50
```

3. Write a program read five numbers store in array and print those five numbers in reverse order.

0	1	2	3	4
10	20	30	40	50

```
void main(  )
{
        int x[5],i;
        clrscr( );
        printf(“ enter 5 eles : ”);
        for(i=0;i<=4;i++)
                scanf(“%d”,&x[i]);
        printf(“eles :”);
        for(i=4;i>=0;i--)
                printf(“ %d \t”,x[i]);
        getch( );
}
```

OUTPUT:

```
enter 5 eles : 10  20  30  40  50
else : 50     40     30    20   10
```

4. Write a program to sort the given elements.

0	1	2	3	4
10	5	3	1	6

```
void main(  )
{
        int x[5],i,j,t;
        clrscr( );
        printf(“ enter 5 eles : ”);
        for(i=0;i<=4;i++)
                scanf(“%d”,&x[i]);
        printf(“before sorting eles : \n”);
        for(i=0;i<=4;i++)
                printf(“ %d \t”,x[i]);
        for(i=0;i<=3;i++)
        {
                for(j=i+1;j<=4;j++)
                {
                        if(x[i]>x[j])
                        {
                                t=x[i];
                                x[i]=x[j];
                                x[j]=t;
                        }
                }
        }
        printf(“after sorting eles :\n”);
        for(i=0;i<=4;i++)
                printf(“ %d \t”,x[i]);
        getch( );
}
```

OUTPUT:

```
enter 5 eles : 10    5    3   1    6
before sorting eles :
10       5      3     1      6

after sorting eles :
1       3      5     6      10
```

5. Write a program to find maximum and minimum number in given array.

0	1	2	3	4
10	5	3	1	6

```
void main( )
{
        int x[5], i, max, min;
        clrscr( );
        printf(“ enter 5 eles : ”);
        for(i=0;i<=4;i++)
                scanf(“%d”,&x[i]);
        max=min=x[0];
        for(i=1;i<=4;i++)
        {
                if(x[i]>max)
                        max=x[i];
                if(x[i]<min)
                        min=x[i];
        }
        printf(“max :%d  min :%d”,max,min);
        getch( );
}
```

OUTPUT:

```
enter 5 eles : 10   5    3    1    6
max  :10   min  : 1
```

2D Array: -

Declaration: -

Data Type   Variable[size][size];

Ex: -

```
        int  x[3][3];
        float a[4][2];
        char s[10][20];
```

1. Simple program for printing 2*2 matrices elements.

```
void main( )
{
        int x[2][2],i,j;
        clrscr( );
        printf(“ enter 4 eles : ”);
        for(i=0;i<=1;i++)
        for(j=0;j<=1;j++)
                scanf(“%d”,&x[i][j]);
        printf(“eles :\n”);
        for(i=0;i<=1;i++)
        {
                for(j=0;j<=1;j++)
                        printf(“ %d \t”,x[i][j]);
                printf(“\n”);
        }
        getch( );
}
```

OUTPUT:

```
enter 4 eles : 1   2   3   4
eles :
1       2
3       4
```

$$\begin{bmatrix}00 & 01\\10 & 11\end{bmatrix} \qquad \begin{bmatrix}11 & 12\\21 & 22\end{bmatrix}$$

2. Simple program for printing 3*3 matrices elements.

```
void main( )
{
        int x[3][3],i,j;
        clrscr( );
        printf(" enter 9 eles : ");
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf("%d",&x[i][j]);
        printf("eles : \n");
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(" %d \t",x[i][j]);
                printf("\n");
        }
        getch( );
}
```

```
OUTPUT:

enter 9 eles : 1   2   3   4  5  6  7  8  9
eles :
1   2   3
4   5   6
7   8   9
```

$$\begin{bmatrix} 00 & 01 & 02 \\ 10 & 11 & 12 \\ 20 & 21 & 22 \end{bmatrix} \qquad \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{bmatrix}$$

3. Write a program to print transpose of the given matrices.

```
void main( )
{
        int x[3][3],i,j;
        clrscr( );
        printf(" enter 9 eles : ");
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf("%d",&x[i][j]);
        printf("eles : \n");
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(" %d \t",x[i][j]);
                printf("\n");
        }
        printf("\n transpose of the given matrices: \n");
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(" %d \t",x[j][i]);
                printf("\n");
        }
        getch( );
}
```

```
OUTPUT:
enter 9 eles :1 2 3 4 5 6 7  8  9
eles :
1    2    3
4    5    6
7    8    9
transpose of the given matrices:
1    4   7
2    5  8
3    6  9
```

$$\begin{bmatrix} 00 & 01 & 02 \\ 10 & 11 & 12 \\ 20 & 21 & 22 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

4. Write a program to print trace of the given matrices.

```
void main(  )
{
        int x[3][3],i,j,sum=0;
        clrscr( );
        printf(" enter 9 eles : ");
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf("%d",&x[i][j]);
        printf("eles :");
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(" %d \t",x[i][j]);
                printf("\n");
        }
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                {
                        if(i = = j)
                                sum = sum + x[i][j];
                }
        }
        printf("\n trace of the given matrices: \n",sum);
        getch( );
}
```

Output:

```
enter 9 eles : 1  2  3  4  5  6   7  8  9
eles:                                0     1    2
1        2        3             0  | 00    01   02 |
4        5        6             1  | 10    11   12 |
7        8        9             2  | 20    21   22 |

trace of the given matrices:
15                                  i            j
                                   □            □
```

5. Write a program to add two matrices.   (3*3) with (3*3)

```
void main(  )
{
        int x[3][3],y[3][3],z[3][3]i,j;
        clrscr( );
        printf(" enter 9 eles in x : ");
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf("%d",&x[i][j]);
        printf("enter 9 eles in y :");
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf("%d",&y[i][j]);
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                z[i][j]= x[i][j] + y[i][j];
        printf(" \n eles in x : \n");
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf("%d\t",x[i][j]);
                printf("\n");
        }
```

Output:

```
enter 9 eles in x : 1  2  3  4  5  6   7  8  9
enter 9 eles in y : 1  1  1  1  1  1   1  1  1

eles in x :                          0     1    2
1        2        3             0  | 00    01   02 |
4        5        6             1  | 10    11   12 |
7        8        9             2  | 20    21   22 |

eles in y :
1        1        1
1        1        1
1        1        1

eles in z :
2        3        4
5        6        7
8        9        10
```

```
        printf(“ \n eles in y : \n”);
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(“%d\t”,y[i][j]);
                printf(“\n”);
        }
        printf(“ \n eles in z : \n”);
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(“%d\t”,z[i][j]);
                printf(“\n”);
        }
        getch( );
}
```

6. Write a program to multiple two matrices.   (3*3) with (3*3)

```
void main(  )
{
        int x[3][3],y[3][3],z[3][3]i,j,k;
        clrscr( );
        printf(“ enter 9 eles in x : ”);
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf(“%d”,&x[i][j]);
        printf(“enter 9 eles in y :”);
        for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
                scanf(“%d”,&y[i][j]);
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                {
                z[i][j]=0;
                for(k=0;k<=2;k++)
                z[i][j]= z[i][j] + (x[i][k] * y[k][j]);
                }
        }
        printf(“ \n eles in x : \n”);
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(“%d\t”,x[i][j]);
                printf(“\n”);
        }
        printf(“ \n eles in y : \n”);
        for(i=0;i<=2;i++)
        {
                for(j=0;j<=2;j++)
                        printf(“%d\t”,y[i][j]);
                printf(“\n”);
        }
```

Output:

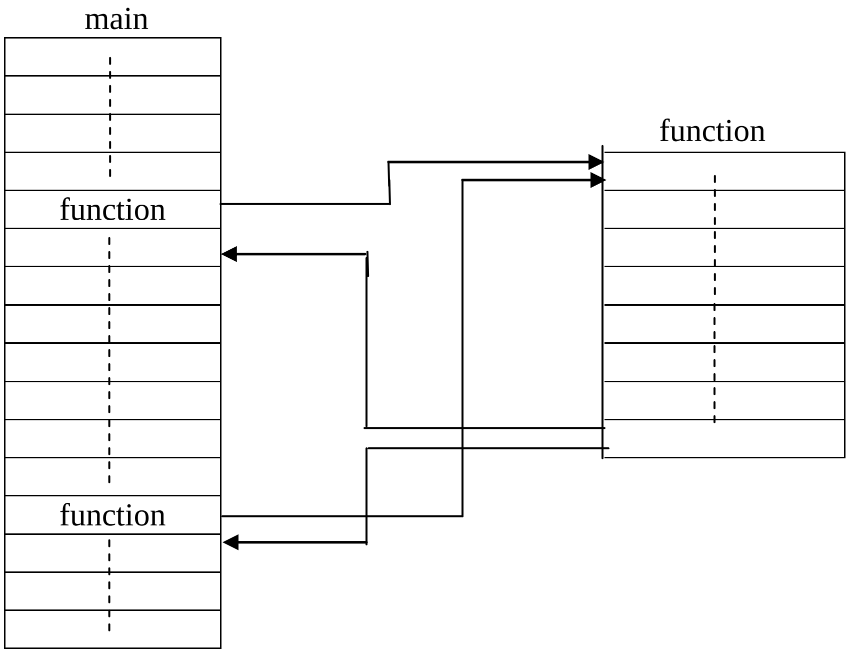
```
enter 9 eles in x : 1  2  3  4  5  6   7  8  9
enter 9 eles in y : 1  1  1  1  1  1   1  1  1

eles in x :                        0     1    2
1         2        3          0  | 00    01   02 |
4         5        6          1  | 10    11   12 |
7         8        9          2  | 20    21   22 |

eles in y :                        0     1    2
1         0         0         0  | 00    01   02 |
0         1         0         1  | 10    11   12 |
0         0         1         2  | 20    21   22 |

eles in z :
1         2         3
4         5         6
7         8         9
```

```
    printf(“ \n eles in z : \n”);
    for(i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
            printf(“%d\t”,z[i][j]);
        printf(“\n”);
    }
    getch( );
}
```

FUNCTIONS

Def: - Sub program.

Function is sub program. It having own name and block of statements. When it is called from main it will be executed.



Types of functions: -

1. Passing by arguments & Return type
2. Passing by arguments & Non-return type
3. Passing by reference & Return type
4. Passing by reference & Non-return type

1. Simple program on function.

```
void main( )                    ↗ function declaration
{
        void function( );
        clrscr( );
        printf("\n we are in main");
        function( );  ──────────────→ calling of function
        printf("\n we are in main");
        getch( );
}
void function( )  ──────────────→ called function
{
        printf("\n we are in function");
}
```

Output:

```
we are in main
we are in function
we are in main
```

2. Simple program on function.

```
void function1( );
void function2( );

void main( )
{
        clrscr( );
        printf("\n we are in main");
        function1( );
        printf("\n we are in main");
        function2( );
        printf("\n we are in main");
        getch( );
}

void function1( )
{
        printf("\n we are in function1");
        function2( );
        printf("\n we are in function1");
}

void function2( )
{
        printf("\n we are in function2");
}
```

Output:

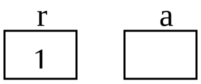
```
we are in main
we are in function1
we are in function2
we are in function1
we are in main
we are in function2
we are in main
```

3. Write a program to print area of the circle. (Using function)

```
void main( )
{
        void area( );
        clrscr( );
        area( );
        getch( );
}

void area( )
{
        int r;
        float a;
        printf("enter r :");
        scanf("%d",&r);
        a= 3.14 * r * r;
        printf("area of the circle :%f",a);
}
```

Output :

```
enter r : 1
area of the circle : 3.140000
```

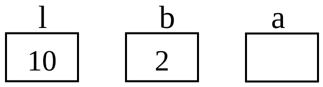


4. Write a program to print area of the rectangle. (Using function)

```
void main( )
{
        void area( );
        clrscr( );
        area( );
        getch( );
}
void area( )
{
        int l,b,a;
        printf("enter l b :");
        scanf("%d%d",&l,&b);
        a= l * b;
        printf("area of the rectangle :%d",a);
}
```

Output :

```
enter l b : 10    2
area of the rectangle : 20
```

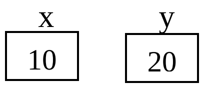


5. Write a program to swapping or interchanging of two numbers. (Using function)

```
void main( )
{
        void swap( );
        clrscr( );
        swap( );
        getch( );
}
void swap( )
{
        int x,y;
        printf("enter two nos :");
        scanf("%d%d",&x,&y);
        printf(" x : %d  y : %d \n",x,y);
        x=x+y;
        y=x-y;
        x=x-y;
        printf(" x : %d  y : %d \n",x,y);
}
```

Output :

```
enter two nos : 10    20
x : 10  y : 20
x : 20  y : 10
```



6. Write a program to print reverse of the given number. (Using function)

```
void main( )
{
        void reverse( );
        clrscr( );
        reverse( );
        getch( );
}
```

Output :

```
enter no : 123
reverse of the given no :321
```

```
void reverse( )
{
        int num,rem,sum=0;                                        num        rem       sum
        printf("enter no :");                                     123                  0
        scanf("%d",&num);
        while(num>0)
        {
                rem = num % 10;
                sum = (sum * 10) + rem;
                num = num / 10;
        }
        printf("reverse of the given no :%d",sum);
}
```

7. Write a program to print area of the circle. (Using passing by argument & non-return type)

```
void main( )
{
        void area( int r );
        int r;                              r
        clrscr( );                          1
        printf("enter r :");
        scanf("%d",&r);
        area( r ); → actual parameter
        getch( );
}
void area( int r ) → formal parameter       r      a
{                                           1
        float a; → local variable
        a= 3.14 * r * r;
        printf("area of the circle :%f",a);
}
```

Output :

```
enter r : 1
area of the circle : 3.140000
```

8. Write a program to print area of the rectangle. (Using passing by argument & non-return type)

```
void main( )
{
        void area( int l, int b );
        int l,b;                          l     b
        clrscr( );                        10    2
        printf("enter l b :");
        scanf("%d%d",&l,&b);
        area( l,b );
        getch( );
}
void area(int l, int b )                  l      b      a
{                                         10     2
        int a;
        a= l * b;
        printf("area of the rectangle :%d",a);
}
```

Output :

```
enter l b : 10   2
area of the rectangle : 20
```

9. Write a program to swapping or interchanging of two numbers. (Using passing by argument & non-return type)

```
void main( )
{
        void swap( int x, int y );
        int a,b;
        clrscr( );
        printf("enter two nos :");
        scanf("%d%d",&a,&b);
        printf(" a : %d  b : %d \n",a,b);
        swap( a,b);
        getch( );
}
void swap( int x, int y )
{
        x=x+y;
        y=x-y;
        x=x-y;
        printf(" a : %d  b : %d \n",x,y);
}
```

Output :

```
enter two nos : 10    20

a : 10  b : 20
a : 20  b : 10
```

```
  a       b
[10]    [20]

  x       y
[10]    [20]
```

10. Write a program to print reverse of the given number. (Using passing by argument & non-return type)

```
void main( )
{
        void reverse( int num );
        int n;
        clrscr( );
        printf("enter no :");
        scanf("%d",&n);
        reverse( n );
        getch( );
}

void reverse(  int num )
{
        int rem,sum=0;
        while(num>0)
        {
                rem = num % 10;
                sum = (sum * 10) + rem;
                num = num / 10;
        }
        printf("reverse of the given no :%d",sum);
}
```

Output :

```
enter no : 123
reverse of the given no :321
```

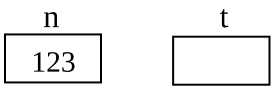
```
   n
[123]

  num      rem      sum
[123]    [   ]    [ 0 ]
```

11. Write a program to print reverse of the given number. (Using passing by argument & return type)

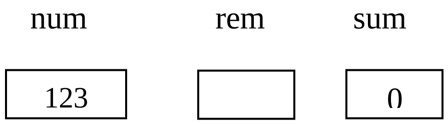
```
void main( )
{
        int reverse( int num );
        int n,t;
        clrscr( );
        printf("enter no :");
        scanf("%d",&n);
        t = reverse( n );
        printf("reverse of the given no :%d",t);
        getch( );
}
```

Output :

```
enter no : 123
reverse of the given no :321
```



```
int reverse(  int num )
{
        int rem,sum=0;
        while(num>0)
        {
                rem = num % 10;
                sum = (sum * 10) + rem;
                num = num / 10;
        }
        return( sum );
}
```

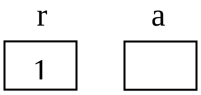


12. Write a program to print area of the circle. (Using passing by argument & return type)

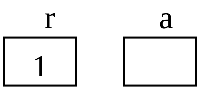
```
void main( )
{
        float area( int r );
        int r;
        float a;
        clrscr( );
        printf("enter r :");
        scanf("%d",&r);
        a = area( r );
        printf("area of the circle :%f",a);
        getch( );
}
```

Output :

```
enter r : 1
area of the circle : 3.140000
```



```
float area( int r )
{
        float a;
        a= 3.14 * r * r;
        return( a );
}
```



13. Program on Local Variable.

```
void main( )                                             OUTPUT:
{
        void  gopi( );                                   x  : 5
        int x=5;                                         x  : 10
        clrscr( );                                       x  : 15
        printf(“x  :%d\n”,x);                            x  : 5
        gopi( );
        printf(“x  :%d\n”,x);
        getch( );
}
void  gopi( )
{
        int  x=10;
        printf(“x  :%d\n”,x);
        x=x+5;
        printf(“x  :%d\n”,x);
}
```

14. Program on Global Variable.

```
int x = 5;  ————→ Global variable
void main( )                                             OUTPUT:
{
        void  gopi( );                                   x  : 5
        clrscr( );                                       x  : 5
        printf(“x  :%d\n”,x);                            x  : 10
        gopi( );                                         x  : 10
        printf(“x  :%d\n”,x);
        getch( );
}
void  gopi( )
{
        printf(“x  :%d\n”,x);
        x=x+5;
        printf(“x  :%d\n”,x);
}
```

15. Write a program add, subtract, multiply, divide and find reminder of two numbers when operator is given (+ - * / %).

```
void main( )                                             Output:
{
        void add(int x, int y); void sub(int x, int y);  enter op [+ - * / %]: +
        void mul(int x, int y); void div(int x, int y);  ente two nos: 10    3
        void rem(int x, int y);                          13
        int a,b;          char op;
        clrscr( );
```

```
          printf(“enter op [+ - * / %] :”);      scanf(“%c”,&op);
          printf(“enter two nos:”);              scanf(“%d%d”,&a,&b);
          switch( op )
          {       case ‘+’ :  add(a,b);  break;
                  case ‘-’ :   sub(a,b);  break;
                  case ‘*’ :  mul(a,b);  break;
                  case ‘/’ :   div(a,b);  break;
                  case ‘%’ : rem(a,b);  break;
                  default : printf(“ enter correct option”);
          }
          gecth();
}
void add(int x, int y)
{         int z;
          z = x + y;
          printf(“%d”,z);
}
void sub(int x, int y)
{         int z;
          z = x - y;
          printf(“%d”,z);
}
void mul(int x, int y)
{         int z;
          z = x * y;
          printf(“%d”,z);
}
void div(int x, int y)
{         float z;
          z = x /(float) y;
          printf(“%f”,z);
}
void rem(int x, int y)
{         int z;
          z = x % y;
          printf(“%d”,z);
}
```

Recursion: -

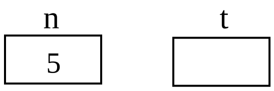
Function calling it self.

1. Write a program to print factorial of the given number. (Using Recursion)

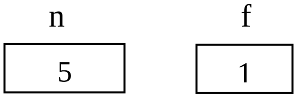
```
void main( )
{
        int fact( int n );
        int n,t;
        clrscr( );
        printf(“enter no :”);
        scanf(“%d”,&n);
        t = fact( n );
        printf(“%d! :%d”,n,t);
        getch( );
}
```

Output :

```
enter no : 5
5! = 120
```



```
int fact(  int n )
{
        int f;
        if(n = =0)
                return(1);
        else
        if(n = = 1)
                return(1);
        else
                f = n * fact (n-1);

        return( f );
}
```



STRINGS

Def: - Collection of characters.

Declaration : -

i) D.T. V[size]=”string”;

ex:-

char x[5]=”gopi”;

0	1	2	3	4
g	o	p	i	‘\0’

char y[10]=”abc”;

0	1	2	3	4	5	6	7	8	9
a	b	c	‘\0’						

ii) D.T. V[size];

ex:-

char x[10];

0	1	2	3	4	5	6	7	8	9

char a[0];

0	1	2	3	4	5	6	7	8	9	- - - - -	19

‘\0’ → NULL Character

1. Simple program on strings.

```
svoid main( )
{
        char  x[10] = “gopi”;
        clrscr( );
        printf(“string :%s”,x);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9
x	g	o	p	i	‘\0’					

Output: string : gopi

2. Simple program on strings.

```
void main( )
{
        char  x[20];
        clrscr( );
        printf(“enter string :”);
        scanf(“%s”,x);
        printf(“string :%s”,x);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9		
x	g	o	p	i	‘\0’							

Output:

```
enter string : gopi
string : gopi
```

3. Simple program on strings.

```
void main( )
{
        char  x[20];
        clrscr( );
        puts(“enter string :”);
        gets(x);
        puts(x);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12		
x	g	o	p	i		k	r	i	s	h	n	a	‘\0’		

Output:

```
enter string :
gopi krishna
gopi krishna
```

String functions: -

1. strlen( )
2. strcpy( )
3. strcmp( )
4. strcat( )

<string.h>

strlen( ): - (string length)

Use: - To find string length

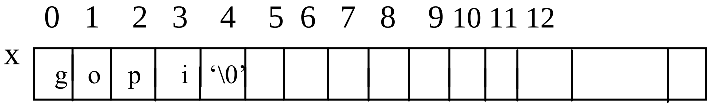
Declaration: -

v = strlen(“string”); or v = strlen(string variable);

```
Ex: -   len     =  strlen(“gopi”);
        a       =  strlen(x);
        b       =  strlen(s);
```

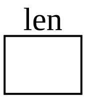
4. Write a program to find string length. [Using string function strlen( )]

```
#include<string.h>
void main( )
{
        char  x[20];
        int len;
        clrscr( );
        printf(“enter string :”);
        scanf(“%s”,x);
        len = strlen(x);
        printf(“ string  :%s \n”,x);
        printf(“ length :%d”,len);
        getch( );
}
```



Output:

```
enter string : gopi
string  : gopi
length : 4
```



strcpy( ): - (string copy)

Use: - To copy string one variable to other.

Declaration: -

strcpy(v,“string”); or strcpy(destination, source);

```
Ex: -   strcpy (s,“gopi”);
        strcpy (x,y);
        strcpy (s1,s2);
```

5. Write a program to copy string one variable to other. [Using string function strcpy( )]

```
#include<string.h>
void main( )
{
        char  x[20],y[20];
        clrscr( );
        printf(“enter string:”);
        scanf(“%s”,x);
        strcpy(y,x);
        printf(“x  :%s\n”,x);
        printf(“y  :%s\n”,y);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	g	o	p	i	‘\0’								

	0	1	2	3	4	5	6	7	8	9	10	11	12
y													

Output:

```
enter string : gopi
x   : gopi
y   : gopi
```

strcmp( ): - (string comparison)

Use: - To compare to strings.

Declaration: -

v = strcmp (s1,s2);

> 0 → s1 > s2

< 0 → s1 < s2

= 0 → s1 = s2

Ex: - a = strcmp (x,y);
b = strcmp (s1,s2);

6. Write a program to compare to strings. [Using string function strcmp( )]

```
#include<string.h>
void main( )
{
        char  x[20],y[20];   int a;
        clrscr( );
        printf(“enter two strings:”);
        scanf(“%s%s”,x,y);
        a = strcmp(x,y);
        if( a  = = 0)
                printf(“ equal ”);
        else
                printf(“ not equal ”);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	g	o	p	i	‘\0’								

	0	1	2	3	4	5	6	7	8	9	10	11	12
	g	o	p	i	‘\0’								

Output:

```
enter two string : gopi    gopi
equal
```

strcat( ): - (string concatenation)

Use: - To combine to strings.

Declaration: -

strcat(s1,s2);

Ex : -

strcat (s1,s2);
strcat (x,y);

7. Write a program to combine two strings. [Using string function strcat( )]

```
#include<string.h>
void main( )
{
        char  x[20],y[20];
        clrscr( );
        printf(“enter two string:”);
        scanf(“%s%s”,x,y);
        printf(“x  :%s\n”,x);
        printf(“y  :%s\n”,y);
        strcat(x,y);
        printf(“x  :%s\n”,x);
        printf(“y  :%s\n”,y);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	g	o	p	i	‘\0’								

0	1	2	3	4	5	6	7	8	9	10	11	12
k	r	i	s	h	n	a	‘\0’					

Output:

```
enter two string : gopi    krishna
x   : gopi
y   : krishna
x   : gopikrishna
y   : krishna
```

strrev( ): - (string reverse)

Use: - To change the reverse of the given string.

Declaration: -

strrev(string variable);

ex: -   strren(s);
strrev(x);
strlen(a);

8. Write a program to print reverse of the given string. [Using string function strrev( )]

```
#include<string.h>
void main( )
{
        char  x[20];
        clrscr( );
        printf(“enter string :”);
        scanf(“%s”,x);
        printf(“ string  :%s \n”,x);
        strrev(x);
        printf(“ string  :%s \n”,x);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	a	b	c	d	‘\0’								

Output:

```
enter string : abcd
string  : abcd
string  : dcba
```

strupr( ) : - (String Upper)

Use :-   To change upper case.

Declaration: -

strupr(sv);

Ex :-

strupr(s);
strupr(x);

strlwr( ) : - (String Lower)

Use :-   To change lower case.

Declaration: -

strlwr(sv);

Ex :-

strlwr(s);
strlwr(x);

9. Write a program to check the given string is palindrome or not.

```
#include<string.h>
void main( )
{
        char  x[20],y[20];
        clrscr( );
        printf(“enter string:”);
        scanf(“%s”,x);
        strcpy(y,x);
        strrev(y);
        if( strcmp(x,y) = = 0)
                printf(“%s is palindrome ”,x);
        else
                printf(“%s is not palindrome ”,x);
        getch( );
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	m	a	d	a	m	‘\0’							

	0	1	2	3	4	5	6	7	8	9	10	11	12
y													

Output:

```
enter string : madam
madam is palindrome
```

10. Write a program to convert lower case string to upper case.

```
#include<ctype.h>
void main( )
{
        char  convert (char  c);
        char  x[20],y[20];
        int  i;
        clrscr( );
        printf(“enter string :”);
        scanf(“%s”,x);
        printf(“string  :%s”,x);
        for(i=0;x[i]!=’\0’;i++)
                y[i]= convert(x[i]);
        y[i]=’\0’;
        printf(“string  :%s”,y);
        getch( );
}

char  convert (char  c)
{
        if( islower(c) )
                return(toupper(c) );
        else
                return(c);
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
x	a	b	c	d	‘\o’								

	0	1	2	3	4	5	6	7	8	9	10	11	12
y													

Output:

```
enter string : abcd
string  : abcd
string  : ABCD
```

STRUCTURES & UNIONS

STRUCTURES

Def: -

Collection of different data types.

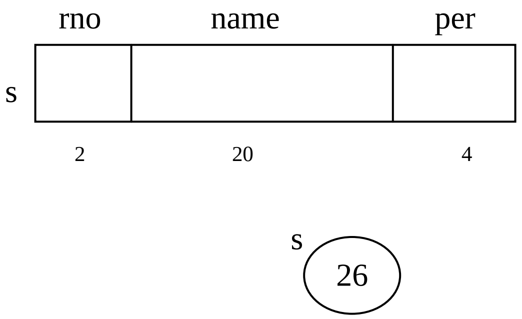
Declaration: -

```
i.  struct  tag
    {
          members of structures;

    };
   struct  tag  variable;
```

```
ii.  struct
     {
          members of structures;

     }variable;
```

1. Write a program to print student details using structures. (i.e. rno, name & per)

```
struct  student
{
        int rno;
        char name[20];
        float per;
};

void main( )
{
        struct student s;
        clrscr( );

        printf("enter rno :");
        scanf("%d",&s.rno);
        printf("enter name :");
        scanf("%s",s.name);
        printf("enter per :");
        scanf("%f",&s.per);

        printf("\n rno        :%d",s.rno);
        printf("\n name     :%s",s.name);
        printf("\n per        :%f",s.per);
        getch( );
}
```

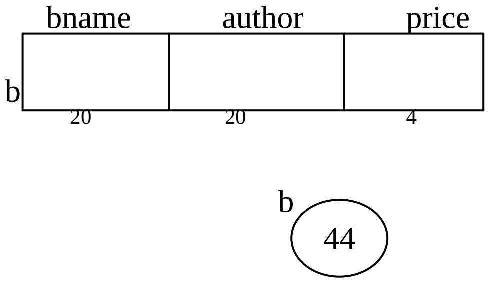


Output:

```
enter rno : 1
enter name : gopi
enter per : 70

rno      : 1
name   : gopi
per      : 70.000000
```

2. Write a program to print book details using structures. (i.e. bname, author & price)

```
struct  book
{
        char bname[20];
        char author[20];
        float price;
};
void main( )
{
        struct book  b;
        clrscr( );
        printf("enter bname :");
        scanf("%s", b.bname);
        printf("enter author :");
        scanf("%s",b.author);
        printf("enter price :");
        scanf("%f",&b.price);
        printf("\n bname   :%s",b.bname);
        printf("\n author    :%s",b.author);
        printf("\n price      :%f",b.price);
        getch( );
}
```

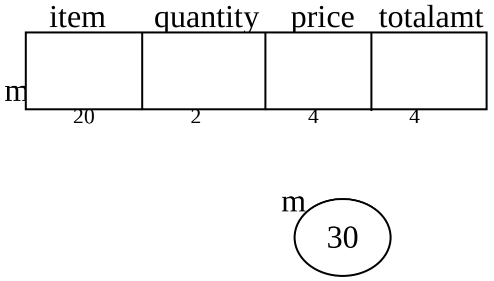


Output:

```
enter bname : c
enter author : swamy
enter price :100

bname     : c
author      : swamy
price       : 100.000000
```

3. Write a program to print purchasing item from super market using structures. (i.e. item, quantity price & totalamt)

```
struct  market
{
        char item[20];
        int quantity;
        float price, totalamt;
};
void main( )
{
        struct market  m;
        clrscr( );
        printf("enter item :");
        scanf("%s", m.item);
        printf("enter quantity :");
        scanf("%d",&m.quantity);
        printf("enter price :");
        scanf("%f",&m.price);
        m.totalamt = m.quantity * m.price;
        printf("\n item          :%s",m.item);
        printf("\n quantity    :%d",m.quantity);
        printf("\n price         :%f",m.price);
        printf("\n totalamt    :%f",m.totalamt);
        getch( );
}
```

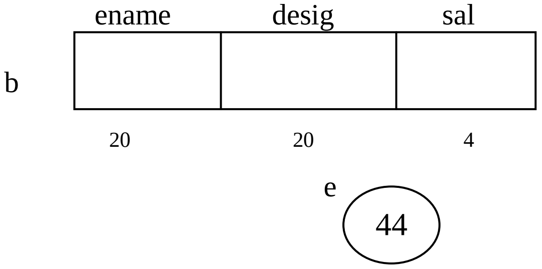


Output:

```
enter item : pen
enter quantity : 5
enter price :10

item            : pen
quantity      : 5
price           : 10.000000
totalamt      :     50.000000
```

4. Write a program to print details of employee using structures. (i.e. ename, desig & sal)

```
struct  employee
{
        char ename[20];
        char desig[20];
        float sal;
};
void main( )
{
        struct employee  e;
        clrscr( );
        printf("enter ename :");
        scanf("%s", e.ename);
        printf("enter desig  :");
        scanf("%s",e.desig);
        printf("enter sal :");
        scanf("%f",&e.sal);
        printf("\n ename   :%s",e.ename);
        printf("\n desig     :%s",e.desig);
        printf("\n sal         :%f",e.sal);
        getch( );
}
```

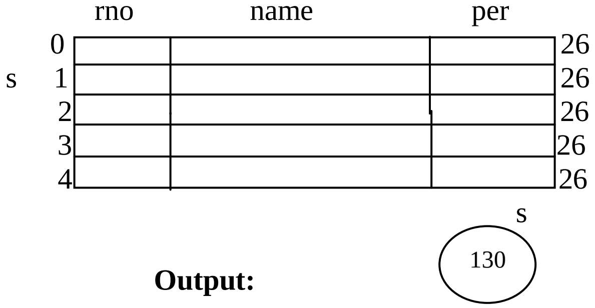


Output:

```
enter ename : ravi
enter desig : jr.asst
enter sal :5000.00

ename     : ravi
desig      : jr.asst
sal          : 5000.000000
```

5. Write a program to print students details using array of structures. (i.e. rno, name & per)

```
struct  student
{
        int rno;
        char name[20];
        float per;
};
void main( )
{       int  i;
        struct student s[5];
        clrscr( );
        for(i=0;i<=4;i++)
        {
                printf("enter rno :");
                scanf("%d",&s[i].rno);
                printf("enter name :");
                scanf("%s",s[i].name);
                printf("enter per :");
                scanf("%f",&s[i].per);
        }
        for(i=0;i<=4;i++)
        {
                printf("\n rno        :%d",s[i].rno);
                printf("\n name     :%s",s[i].name);
                printf("\n per       :%f",s[i].per);
        }
        getch( );
}
```

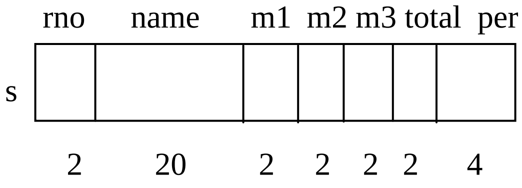


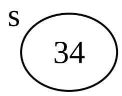
Output:

```
enter rno : 1
enter name : gopi
enter per : 70
          :
          :
enter rno : 5
enter name : xyz
enter per : 60

rno      : 1
name    : gopi
per        : 70.000000
          :
rno      : 5
name    : xyz
per        : 60.000000
```

6. Write a program to print student details using passing an argument as a structures. (i.e. rno, name, m1, m2, m3, total & per)

```
struct  student
{
        int rno;
        char name[20];
        int m1,m2,m3,total;
        float per;
};
void main( )
{
        struct student s;
        void  print(struct student s);
        clrscr( );
        printf(“enter rno :”);
        scanf(“%d”,&s.rno);
        printf(“enter name :”);
        scanf(“%s”,s.name);
        printf(“enter three sub marks :”);
        scanf(“%d%d%d”,&s.m1,&s.m2,&s.m3);
        print(s);
        getch( );
}

void  print(struct student s)
{
        s.total =  s.m1 + s.m2 + s.m3;
        s.per = s.total / 3.0;
        printf(“\n rno        :%d”,s.rno);
        printf(“\n name     :%s”,s.name);
        printf(“\n m1        :%d”,s.m1);
        printf(“\n m2        :%d”,s.m2);
        printf(“\n m3        :%d”,s.m3);
        printf(“\n total      :%d”,s.total);
        printf(“\n per       :%f”,s.per);
}
```



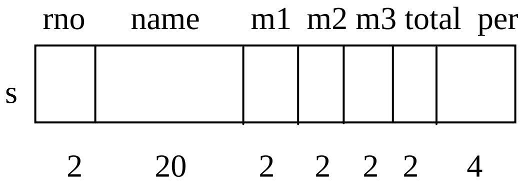


Output:

```
enter rno : 1
enter name : gopi
enter three sub marks : 70    70   70

rno       : 1
name    : gopi
m1        : 70
m2        : 70
m3        : 70
total     : 210
per       : 70.000000
```

7. Write a program to print student details using passing an argument as a structures & return type. (i.e. rno, name, m1, m2, m3, total & per)

```
struct  student
{
        int rno;
        char name[20];
        int m1,m2,m3,total;
        float per;
};
```

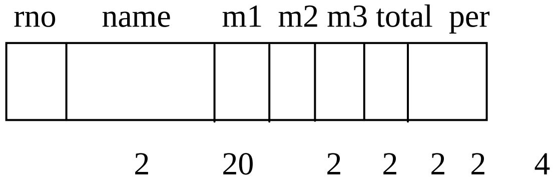


```
void main( )
{
        struct student s;
        struct  student  print(struct student s);
        clrscr( );
        printf("enter rno :");
        scanf("%d",&s.rno);
        printf("enter name :");
        scanf("%s",s.name);
        printf("enter three sub marks :");
        scanf("%d%d%d",&s.m1,&s,m2,&s.m3);
        s = print(s);
        printf("\n rno        :%d",s.rno);
        printf("\n name       :%s",s.name);
        printf("\n m1         :%d",s.m1);
        printf("\n m2         :%d",s.m2);
        printf("\n m3         :%d",s.m3);
        printf("\n total      :%d",s.total);
        printf("\n per        :%f",s.per);
        getch( );
}
```

Output:

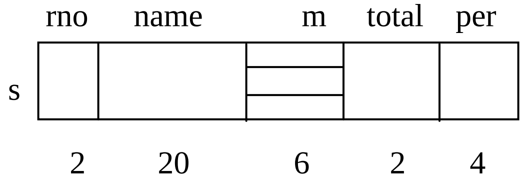
```
enter rno : 1
enter name : gopi
enter three sub marks : 70    70   70

rno      : 1
name     : gopi
m1       : 70
m2       : 70
m3       : 70
total    : 210
per      : 70.000000
```

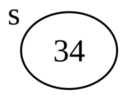
```
struct  student   print(struct student s)
{
        s.total =  s.m1 + s.m2 + s.m3;
        s.per = s.total / 3.0;
        return(s);
}
```



8. Write a program to print student details using array in structures.

```
struct  student
{
        int rno;
        char name[20];
        int m[3],total;
        float per;
};
void main( )
{
        struct student s;
        int  i;
        clrscr( );
        printf("enter rno :");
        scanf("%d",&s.rno);
        printf("enter name :");
        scanf("%s",s.name);
```





Output:

```
enter rno : 1
enter name : gopi
enter three sub marks : 70    70   70
```

```
        printf(“enter three sub marks :”);
        for(i=0;i<=2;i++)
                scanf(“%d”,&s.m[i]);                rno      : 1
        s.total = 0;                                name     : gopi
        for(i=0;i<=2;i++)                           m1       : 70
                s.total =  s.total + s.m[i];        m2       : 70
        s.per = s.total / 3.0;                      m3       : 70
        printf(“\n rno       :%d”,s.rno);           total    : 210
        printf(“\n name      :%s”,s.name);          per      : 70.000000
        for(i=0;i<=2;i++)
                printf(“\n m[%d]       :%d”,i,s.m[i]);
        printf(“\n total     :%f”,s.total);
        printf(“\n per       :%f”,s.per);
}
```

Nested Structure : -

Structure within structure.

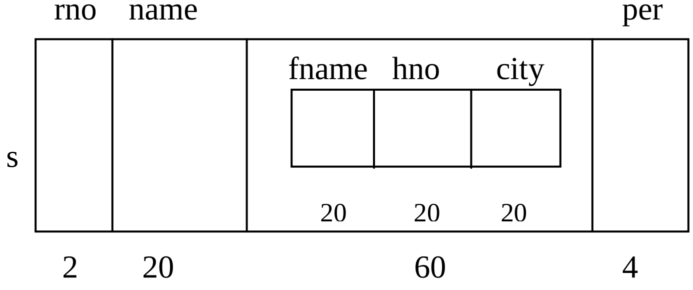
Declaration: -

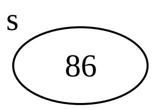
```
struct  tag1
{
        ------
        ------
        struct  tag2
        {
         ┌──────────────────────────┐
         │  members of structures;  │
         └──────────────────────────┘
        } variable2;
} variable1;
```

```
struct  tag1
{
   ┌──────────────────────────┐
   │ members of structures;   │
   └──────────────────────────┘
};
struct  tag2
{
  ┌────────────────────────────┐
  │  members of structures;    │
  │                            │
  │  struct  tag1  variable1   │
  └────────────────────────────┘
}variable2;
```

8. Write a program to print student details using nested structures.

```
struct  address
{
        char  fname[20];
        char  hno[20];
        char city[20];
};

struct  student
{
        int  rno;
        char  name[20];
        struct  address  a;
        float  per;
};
```





```
void  main( )
{
        struct  student  s;
        clrscr( );
        printf(“enter rno      :”);
        scanf(“%d”,&s.rno);
        printf(“enter name  :”);
        scanf(“%s”,s.name);
        printf(“enter fname :”);
        scanf(“%s”,s.a.fname);
        printf(“enter hno     :”);
        scanf(“%s”,s.a.hno);
        printf(“enter city     :”);
        scanf(“%s”,s.a.city);
        printf(“enter per     :”);
        scanf(“%f”,&s.per);

        printf(“\n  rno        :%d”,s.rno);
        printf(“\n  name     :%s”,s.name);
        printf(“\n  fname    :%s”,s.a.fname);
        printf(“\n  hno       :%s”,s.a.hno);
        printf(“\n  city       :%s”,s.a.city);
        printf(“\n  per        :%f”,s.per);

        getch( );
}
```

Output:

```
enter rno        : 1
enter name    : ravi
enter fname   : ram
enter hno       : 1-2-34/A/2
enter city       : Hyd
enter per        : 70

rno            : 1
name         : ravi
fname        : ram
hno           : 1-2-34/A/2
city           : hyd
per            : 70.000000
```

UNIONS

Def: -

Collection of different data types.

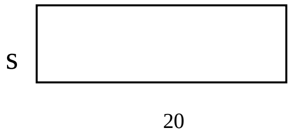
Declaration: -

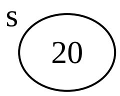
```
i.  union  tag
    {
         members of structures;
    };
   union  tag  variable;
```

```
ii.  union
     {
          members of structures;
     }variable;
```

1. Write a program to print student details using union. (i.e. rno, name & per)

```
union  student
{
        int rno;
        char name[20];
        float per;
};

void main( )
{
        union  student  s;
        clrscr( );

        printf(“enter rno :”);
        scanf(“%d”,&s.rno);
        printf(“ rno            :%d \n”,s.rno);
        printf(“enter name :”);
        scanf(“%s”,s.name);
        printf(“\n name      :%s \n”,s.name);
        printf(“enter per :”);
        scanf(“%f”,&s.per);
        printf(“\n per         :%f”,s.per);
        getch( );
}
```





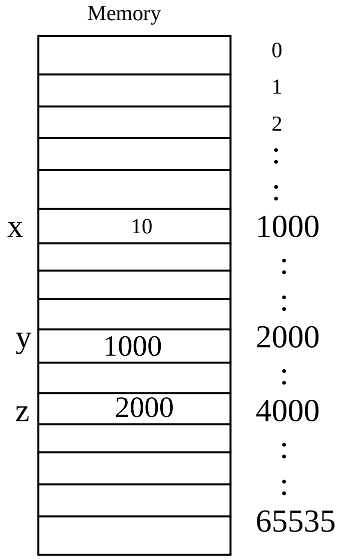
Output:

```
enter rno : 1
rno       : 1
enter name  : gopi
name    : gopi
enter per : 70
per        : 70.000000
```

POINTERS

Def: -

Address specifier.

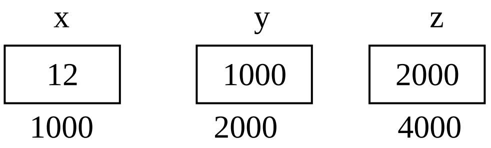


Declaration: -

Data type *variable;

Ex: -

```
int  *p;
char  *k;
float  *a;
```



```
x    : 10        y    : 1000     z    : 2000
&x   : 1000      &y   : 2000     &z   : 4000
```

1. Simple program on pointers.

```
void  main( )
{
        int  x=10,*y;
        clrscr( );
        y=&x;
        printf("x    :%d   &x  :%u \n",x,&x);
        printf("y    :%u   &y  :%u \n",y,&y);
        printf("*y  :%d ",*y);
        getch( );
}
```

Output:

```
x   :10          &x   :1000
y   :1000        &y   :2000
*y :10
```

2. Simple program on pointers.

```
void  main( )
{
        int  x=10,*y,**z;
        clrscr( );
        y = &x;             z = &y;
        printf("x    :%d   &x  :%u \n",x,&x);
        printf("y    :%u   &y  :%u \n",y,&y);
        printf("z    :%u   &z  :%u \n",z,&z);
        printf("*y  :%d   *z   :%u \n",*y,*z);
        printf("**z:%d",**z);
        getch( );
}
```

Output:

```
x   :10          &x   :1000
y   :1000        &y   :2000
z   :2000        &z   :4000
*y :10           *z   :1000
**z:10
```

3. Simple program of pointer.

	0	1	2	3	4
×	10	20	30	40	50
	1000	1002	1004	1006	1008

```
void main( )
{
        int   x[5]={10,20,30,40,50};
        clrscr( );
        printf("x[0]  :%d    &x[0]  :%u \n",x[0],&x[0]);
        printf("x[1]  :%d    &x[1]  :%u \n",x[1],&x[1]);
        printf("x[2]  :%d    &x[2]  :%u \n",x[2],&x[2]);
        printf("x[3]  :%d    &x[3]  :%u \n",x[3],&x[3]);
        printf("x[4]  :%d    &x[4]  :%u \n",x[4],&x[4]);
        getch( );
}
```

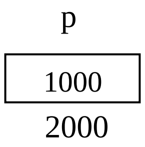
Output:

```
x[0]  :10        &x[0]   :1000
x[1]  :20        &x[1]   :1002
x[2]  :30        &x[2]   :1004
x[3]  :40        &x[3]   :1006
x[4]  :50        &x[4]   :1008
```

4. Write a program to print array of element using pointer.

	0	1	2	3	4
×	10	20	30	40	50
	1000	1002	1004	1006	1008

```
void main( )
{
        int   x[5]={10,20,30,40,50},*p,i;
        clrscr( );
        p=&x[0];
        for(i=0;i<=4;i++)
        {
                printf("%d \n",*p);
                p++;
        }
        getch( );
}
```



Output:

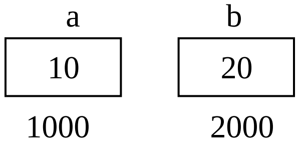
```
10
20
30
40
50
```

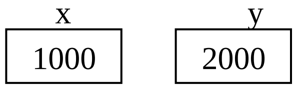
5. Write a program to swap two numbers using passing by reference.

```
void   main( )
{
        void  swap(int  *x,int  *y);
        int  a,b;
        clrscr( );
        printf("enter  two  nos:");
        scanf("%d%d",&a,&b);
        printf("a  :%d   b   :%d \n",a,b);
        swap(&a,&b);
        printf("a  :%d   b   :%d \n",a,b);
        getch( );
}
void  swap(int  *x,int  *y)
{
        *x  =  *x  +  *y;
        *y  =  *x  -  *y;
        *x  =  *x  -  *y;
}
```

Output:

```
enter  two  nos:  10   20
a   :10    b  :20
a   :20    b  :10
```





6. Write a program to find length of the given string. (using passing by reference)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
×	a	b	c	d	‘\0’										
	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1001	1002	1003	1004

```
void main( )
{
        void length(char  *x);
        char  x[15];
        clrscr( );
        printf(“enter  string :”);
        scanf(“%s”,x);
        length( x );
        getch( );
}
void length(char  *x)
{
        int  len = 0;
        while( *x != ‘\0’ )
        {
                printf(“%c”,*x);
                x++;
                len++;
        }
        printf(“ length :%d”,len);
}
```

x	len
1000	0

Output:

```
enter  string : abcd
abcd
length : 4
```

7. Write a program to print student details using pointers. (i.e. rno, name & per)

```
struct  student
{
        int rno;
        char name[20];
        float per;
};
void main( )
{
        struct student s,*p;
        clrscr( );
        printf(“enter rno :”);
        scanf(“%d”,&s.rno);
        printf(“enter name :”);
        scanf(“%s”,s.name);
        printf(“enter per :”);
        scanf(“%f”,&s.per);
        p=&s;
        printf(“\n rno        :%d”,p→rno);
        printf(“\n name     :%s”, p→name);
        printf(“\n per        :%f”, p→per);
        getch( );
}
```

	rno	name	per
s			
	1000	1002	1022

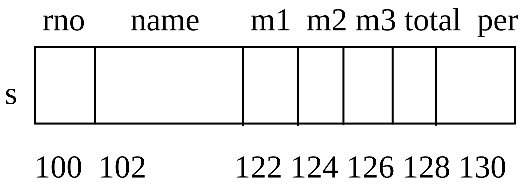
p
1000

Output:

```
enter rno : 1
enter name : gopi
enter per : 70

rno       : 1
name    : gopi
per       : 70.000000
```

8. Write a program to print student details using passing by reference as a structures. (i.e. rno, name, m1, m2, m3, total & per)

```
struct  student
{
        int rno;
        char name[20];
        int m1,m2,m3,total;
        float per;
};
```

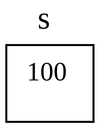


```
void main( )
{
        struct student s;
        void   print(struct  student  *s);
        clrscr( );
        printf("enter rno :");
        scanf("%d",&s.rno);
        printf("enter name :");
        scanf("%s",s.name);
        printf("enter three sub marks :");
        scanf("%d%d%d",&s.m1,&s,m2,&s.m3);
        print(&s);
        printf("\n rno        :%d",s.rno);
        printf("\n name     :%s",s.name);
        printf("\n m1        :%d",s.m1);
        printf("\n m2        :%d",s.m2);
        printf("\n m3        :%d",s.m3);
        printf("\n total      :%d",s.total);
        printf("\n per       :%f",s.per);
        getch( );
}
```

Output:

```
enter rno : 1
enter name : gopi
enter three sub marks : 70    70   70

rno       : 1
name     : gopi
m1        : 70
m2        : 70
m3        : 70
total     : 210
per       : 70.000000
```

```
void  print(struct  student  *s)
{
        s→total =  s→m1 + s→m2 + s→m3;
        s→per = s→total / 3.0;
}
```



FILES

Def: -

Collection of information or collection of records.

Declaration: -

```
FILE  *p;
```

File functions: -

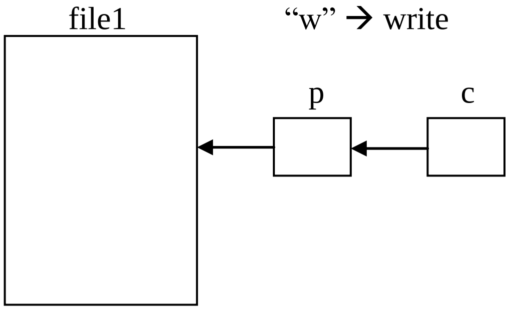
fopen( )
fclose( )
fputc( )
fgetc( )
putw( )
getw( )
fputs( )
fgets( )
fprintf( )
fscanf( )

1. Write a program to create a file. or write information into the file.(using files)

```
#include<stdio.h>
void main( )
{
        FILE  *p;
        char   c;
        clrscr( );
        p = fopen(“file1”,”w”);
        printf(“enter text (at last press ctrl + z) :”);

        while (  ( c=getchar( ) )!=EOF)
                putc( c, p );

        fclose(p);
}
```

Output:

```
enter text (at last press ctrl+z): abcdefgh^z
```

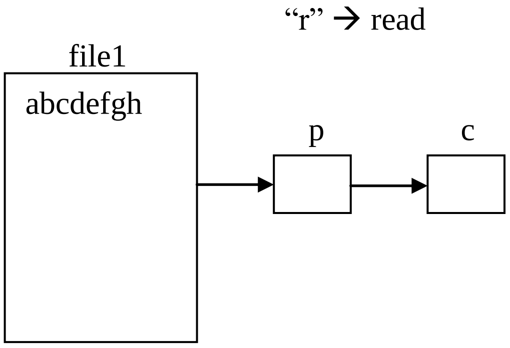


2. Write a program to read information from the file. (using files)

```
#include<stdio.h>
void main( )
{
        FILE  *p;
        char   c;
        clrscr( );
        p = fopen(“file1”,”r”);
        printf(“text :”);

        while (  ( c=getc( p ) )!=EOF)
                putchar( c );
        fclose(p);
        getch( );
}
```

Output:

```
text : abcdefgh
```

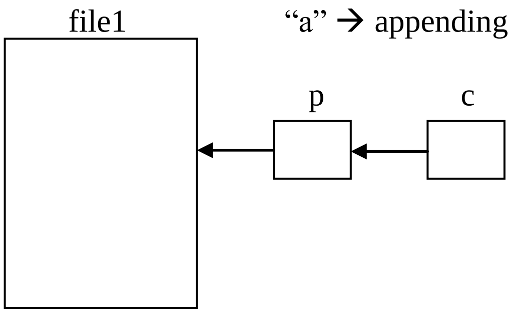


3. Write a program to append or add information to the given file.(using files)

```
#include<stdio.h>
void main( )
{
        FILE  *p;
        char   c;
        clrscr( );
        p = fopen(“file1”,”a”);
        printf(“enter text (at last press ctrl + z) :”);
        while (  ( c=getchar( ) )!=EOF)
                putc( c, p );
        fclose(p);
}
```

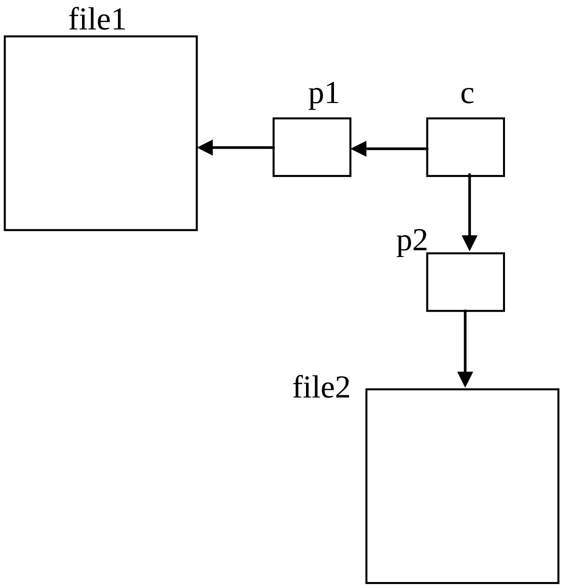
Output:

```
enter text (at last press ctrl+z): abcdefgh^z
```



4. Write a program to copy one file information to other. (using fputc & fgetc)

```
#include<stdio.h>
void main( )
{
        FILE  *p1,*p2;
        char   c;
        clrscr( );
        p1 = fopen(“file1”,”w”);
        printf(“enter text (at last press ctrl + z) :”);
        while (  ( c=getchar( ) )!=EOF)
                putc( c, p1 );
        fclose(p1);

        p1 = fopen(“file1”,”r”);          /* copy */
        p2 = fopen(“file2”,”w”);
        while (  ( c=fgetc( p1 ) )!=EOF)
                fputc( c, p2 );
        fclose( p1 );
        fclose( p2 );

        p1 = fopen(“file1”,”r”);          /* text in file1 */
        printf(“\n text in first file : \n”);
        while (  ( c=getc( p1 ) )!=EOF)
                putchar( c );
        fclose(p1);

        p2 = fopen(“file2”,”r”);          /* text in file2 */
        printf(“\n text in second file : \n”);
        while (  ( c=getc( p2 ) )!=EOF)
                putchar( c );
        fclose(p2);
        getch( );
}
```

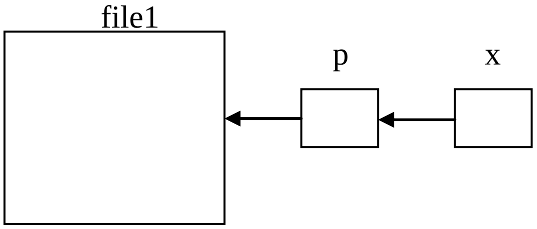


Output:

```
enter text (at last press ctrl+z):abcdefgh^z

text in first file :
abcdefgh

text in second file :
abcdefgh
```

5. Write a program to create file, send strings into the file and read strings from the file. (using fputs & fgets)

```
#include<stdio.h>
void main( )
{
        FILE  *p;
        char   x[20];
        int  i;
        clrscr( );
        p = fopen(“file1”,”w”);
        for(i=1;i<=5;i++)
        {
                printf(“enter string :”);
                scanf(“%s”,x);
                fputs( x , p );
                fputc( ‘\n’, p);
        }
        fclose(p);
        p = fopen(“file1”,”r”);
        printf(“\n strings : \n”);
        while( ( fgets( x, 20, p ) ) != EOF)
                printf(“ %s \n”, x);
        fclose( p );
        getch( );
}
```

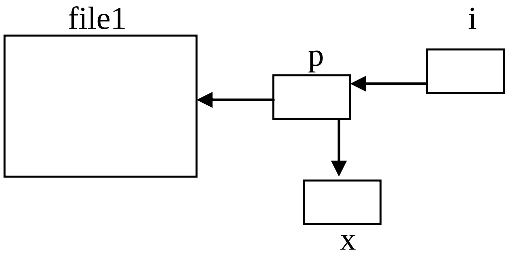


Output:

```
enter string : aaaa
enter string : bbbb
enter string : cccc
enter string : dddd
enter string : eeee

strings :

aaaa
bbbb
cccc
dddd
```

6. Write a program to create a file, send numbers in to the file and read numbers from the file (using putw & getw)

```
#include<stdio.h>
void main( )
{
        FILE  *p;
        int   i,x;
        clrscr( );
        p = fopen(“file1”,”w”);
        for(i=1; i<=10; i++)
                putw( i , p );
        fclose(p);
        p = fopen(“file1”,”r”);
        for(i=1; i<=10; i++)
        {
                x=getw( p );
                printf(“%d \n”, x);
        }
        fclose(p);
        getch( );
}
```

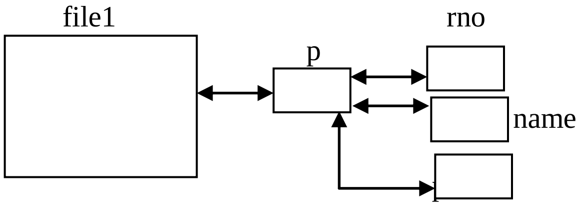


Output:

```
  1
  2
  3
  4
   :
 10
```

7. Write a program to create a file, send student details (rno, name & per) in to the file and read student details from the file. (using fprintf & fscanf)

```
#include<stdio.h>
void main( )
{
          FILE  *p;
          int   rno;
          char name[20];
          float per;
          clrscr( );
          p = fopen(“file1”,”w”);
          printf(“enter rno :”);      scanf(“%d”,&rno);
          printf(“enter name :”);   scanf(“%s”,name);
          printf(“enter per :”);      scanf(“%f”,&per);
          fprintf(p,”%d\t%s\t%f”,rno,name,per);
          fclose(p);
          p = fopen(“file1”,”r”);
          fscanf(p,”%d\t%s\t%f”,&rno,name,&per);
          printf(“\n rno     :%d”,rno);
          printf(“\n name  :%s”,name);
          printf(“\n per     :%f”,per);
          fclose(p);
          getch( );
}
```

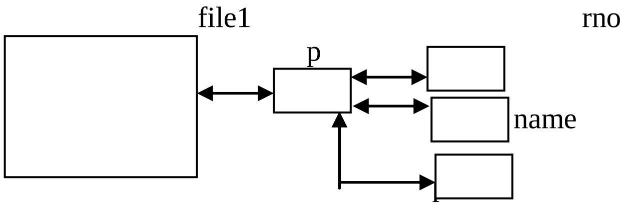


Output:

```
enter rno : 1
          enter name : gopi
          enter per :70

          rno : 1
          name : gopi
          per : 70.000000
```

8. Write a program to create a file, send students details (rno, name & per) in to the file and read student details from the file. (using fprintf & fscanf)

```
#include<stdio.h>
void main( )
{
                    FILE  *p;
          int   rno,i;
          char name[20];
          float per;
          clrscr( );
          p = fopen(“file1”,”w”);
          for(i=1;i<=5;i++)
          {
                    printf(“enter rno :”);      scanf(“%d”,&rno);
                    printf(“enter name :”);   scanf(“%s”,name);
                    printf(“enter per :”);      scanf(“%f”,&per);
                    fprintf(p,”%d\t%s\t%f\n”,rno,name,per);
          }
          fclose(p);
          p = fopen(“file1”,”r”);
          while( fscanf(p,”%d\t%s\t%f\n”,&rno,name,&per)!=EOF)
          {
                    printf(“\n rno     :%d”,rno);
                    printf(“\n name  :%s”,name);
                    printf(“\n per     :%f”,per);
          }
          fclose(p);
          getch( );
}
```



Output:

```
enter rno : 1
enter name : aa
enter per :70
         :
enter rno  :5
enter  name :ee
enter per : 50

rno : 1
name : aa
per : 70.000000
         :
rno : 5
name : ee
per :50.000000
```