```
else {
~~EPP~~
Printf ("Error: unable to rename the file");
}
 getch();
}
```

```
output
File renamed successfully
```

file is
ename
use

(9) fprintf() and fscanf() [Reading and writing to file using)

```
#include <stdio>
struct emp
{
   char name[10];
     int age;
};

void main()
{
  struct emp e;
  FILE *p,*q;

  p= fopen ("ksc.txt", "a");
  q = fopen("ksc.txt","r");
```

```
printf ("Enter Name and Age:");
scanf ("%s %d", e.name, & e.age);

fprintf(p, "%s %d", e.name, e.age);

~~fcdo~~
fclose (p);
do
{
 fscanf (q,"%s %d", e.name, e.age);
  printf ("%s %d", e.name, e.age);
} while (!feof (q));      getch();
}
```

OUTPUT ->

```
Enter Name and Age: 12
Pritam
```

This add to txt.file but only scanf = 12 and Pritam

(10) Dynamic memory Allocation [1] malloc()

```
#include <stdio.h>
#include <conio.h>
int main(){

   int n,i,*ptr,sum=0;
Printf ("Enter number of elements:");-
Scanf ("%d",&n);

Ptr =(int*) malloc (n*sizeof(int)); //2021 2022 2023
//memory allocated using. malloc

if (ptr==Null)
{
 Printf("Sorry! unable to allocate memory");
 exit (0);
}
printf ("Enter elements of array:");
for (i=0; i<n; ++i) // 0 1 2
{
  scanf ("%d", ptr+i);
  sum += (ptr+i);
}
printf ("sum= %d", sum);
free (ptr);
getch();
return 0;
}
```

output=>

```
Enter number of elements: 3
Enter ~~number~~ elements of array: 10
15
20
sum = 45
```

11 calloc()

```
#include<stdio.h>   //memory allocated using calloc
#include<stdlib.h>

int main() {
 int n,i,*ptr, sum=0;
 printf("Enter number of elements:");
 Scanf ("%d",&n);
 Ptr =(int*) calloc(n,sizeof(int));
 //memory allocated using calloc
 if (ptr==NULL)
 {
  Printf("Sorry! unable to allocate memory!");
  exit(0);
 }
 Printf("Enter elements of array:");
 for (i=0;i<n;++i)
 {
   scanf("%d",ptr+i);
   sum += *(ptr+i);
 }
  Printf("sum=%d",sum);
  free(ptr);
  getch();
  return 0;
}
```

output

```
Enter number of elements:3
Enter elements of array :10
15
20
sum=45
```

12 realloc

```
#indude<stdio.h>
#includo<stdib.h>
int main()
{
  // This pointer will hold the
  // base address of the block created
  int *ptr;
  int n,i;
  clrscr();
  //Get the number of elements for the array
  n=5;
  Printf("Enter number~~s~~ of elements :%d\n",n);
  //Dynamically allocate memory using calloc()
  ptr =(int*)calloc (n,sizeof (int));
  //check if the memory has been successfully
  //allocated by Malloc or not

  if (ptr==Null){
     printf("memory not allocated:\n");
     exit(0);
  }
  else {

  //memory has been successfully allocated

  Printf("memory successfully allocated using
        callon.\n");

  //Get the element of array
```

```
for (i=0;i<n;++i){
   ptr[i]=I+1;                //1 2 3 4 5
}

//Print the elements of the array

Printf("The elements of the array are:");
for (i=0;I<n;++i){
   printf("%d,",ptr[i]);
}
//Get the new size for the array
n=10;
Printf("\n\nEnter the new size of the
        array:%d\n",n);

//Dynamically re-allocate memory using
// realloc()

ptr=realloc(ptr,n* size of (int));

//memory has been successfully allocated

printf("Memory successfully ~~re~~-allocated
      using realloc.\n");

//Get the new elements of the array
for (i=5;i<n;++i){
   ptr[i]=i+1; //6 7 8 9 10
}
```

```
//Print the elements of the array

Printf ("The elements of the array are:");

for (i=0; i<n; ++i){
  printf ("%d,", ptr[i]);
 }
 free(ptr);
}
getch();
return 0;
}
```

Output

```
Enter number of elements : 5
Memory successfully allocated using calloc
The elements of the array are: 1,2,3,4,5,

Enter the new size of the array: 10
Memory successfully re-allocated using realloc.
The elements of the array are: 1,2,3,4,5,6,7,8,9,10,
```

13 Ceil() (is return is double)

```
#include <stdio.h>
# include<math.h>

int main(){
    float val1, val2, val3, val4;

val1= 1.6;
val2= 1.2;
val3= 2.8;
val4= 2.3;

printf ("value1 = %.1lf\n", ceil (val1));
Printf ("value2 = %.1lf\n", ceil (val2));
Printf ("value3 =%.1lf\n", ceil (val3));
Printf ("value4 =%.1lf\n", ceil (val4));
getch();
return 0;
}
```

output

```
Value1 = 2.0
Value2 = 2.0
Value3 = 3.0
Value4 = 3.0
```

(14) Pow () x^y

```
#include <stdio.h>
#include <math.h>
int main (){
    clrscr();
   printf ("Value 2.0^3 = %lf\n", pow (2.0, 3));
   printf ("Value 3.05 ^ 1.98 = %lf", pow (3.05, 1.98)
          );
    getch ();
   return 0;
}
```

Output

```
Value  2.0 ^3 = 8.000000
Value  3.65 ^1.98 = 9.097324
```

(15) fabs ()

```
#include<stdio.h>
# include<math.h>

 int main(){
   int a,b;
   a=1234;
   b=-344;

Printf (" The absolute value of %d is %lf \n",
        a, fabls (a));

Printf (" The absolute value of %d is %lf \n",
        b, fabls (b));
 getch ();

 return 0;

 }
```

Output

```
The absolute value of 1234 is 1234.000000
The absolute value of -344 is 344.000000
```

(16) sqrt()

```
#include<stdio.h>
#include< match>

int main(){

  Printf("square root of is %lf \n", 4.0, sqrt(4.0));
  Printf (" square root of is %lf \n", 5.0 ; sqrt(5.0));

  getch();
  return (0);
}
```

Output

```
Square root of  4.000000 is  2.000000
Square root of  5.000000 is  2.236068
```

(17) floor

```
#include<stdio.h>
#include<math.h>
int main (){
  float val1, val2, val3, val4;
  clrscr();

  Val1 = 1.6;
  Val2 = 1.2;
  Val3 = 2.8;
  Val4 = 2.3;
```

PAGE NO.
DATE

```
printf ("value1 = %.1lf\n", fbor (val1));
printf ("value 2 = %.1lf \n", floor (val2));

printf ("floor value3 = %.1lf\n", floor (val3));
Printf ("ceil     value 3 = %.1lf\n", ceil (val3));

Printf ("value4 = %.1lf\n", floor (val4));
 getch ();
 return 0;
}
```

output

```
value1 = 1.0
value2 = 1.0
floor value3 = 2.0
~~floor~~ ceil value3 = 3.0
value4 = 2.0
```

(18) fmod( )

```
#include<stdio.h>
#include<math.h>

int main(){
float a,b;
 int c;

clrscr();

a = 9.2;
b = 3.7;
c = 2;
```

PAGE NO.
DATE

```
printf ("Remainder of %.f/%.d is %.1f\n", a,c,
        fmod (a,c));
printf (" Remainder of %.f / %.f is %.1f \n", a,b
       , fmod (a,b));
   getch ();
  return 0;
}
```

output :-

```
Remaider of 9.2/2 is 1.2
Remaider of 9.2/3.700000 is 1.800000
```

(19) log()

```
#include <stdio.h>
#include<math.h>

 int main () {

  double x,ret;
   clrscr();
   x = 2.7 ;
   // finding log (2.7)
  ret = log(x);
  printf (" log (%.lf) = %.lf", x, ret);
    getch();
  return 0;
 }
```

output

```
log (2.700000) = 0.993252
```

(20) getc()

```
#include<stdio.h>
int main()
{
   printf ("%c", getc(stdin));
   getch ();
  return(0);
}
```

Output

```
r
r
```

(21) getchar()

```
#include<stdio.h>
int main()
{
   clrscr();
   printf ("%c", getchar());
   getch ();
  return 0;
}
```

Output

```
r
r
```

(22) Putc();

```
#include<stdio.h>
void main()
{
char  ch;
FILE  *fp;
if (fp= fopen ("ksc.txt", "r"))
  {
     ch=getc(fp)
    while (ch!=EOF)
    {
     putc(ch, stdout);
      ch = getc(fp);
    }
    fclose(fp);
  }
  getch();
}
```

Output

→ file ksc.txt is wring is i·vier

```
Pritam.
```

- Putchar()

```
// c Program  to demonstrate putchar() method
#include <stdio.h>
void main()
{
 // set  the character to be written
  char ch='G';

  // write the character to stdout
  Putchar (ch);
   getch();
}
```

output

```
GfG
```

- cos & sin

```
#include <stdio.h>
#include <math.h>
void main()
{
 float i= 0.314;
 float j= 0.25;

 float sin_value = sin(i);
 float cos_value = cos (j);

Printf("\n\n i=SIN =%f", sin_value);
Printf("\n\n j= COS =%f", cos_value);

  getch();
}
```

Output

```
SIN = 0.308866
COS = 0.968912
```

- isalpha()

```
#include <stdio.h>
#include <ctype.h>
```

```
Void main(){

char c;

clrscr();

c='Q';
printf("\n Result when uppercase alphabet is
          passed:%d", isalpha(c));

c='q';
printf("\n Result when lowercase alphabet is
          passed:%d", isalpha(c));

c="+";
printf("\n Result when non-alphabetic charater
        is passed:%d", isalpha(c));
  getch();
}
```

output

```
Result when uppercase alphabet is passed:4
Result when lowercase alphabet is passed:8
Result when non-alphabetic character is
passed:0
```

- iscntru ();

```
#include <stdio.h>
#include <ctype.h>

void main()
{
  char c;
  int result;

   clrscr();
  c = "Q'
  result = iscntrl (c);
  printf ("when %c is passed to iscntrl() = %d\n",
          c result);

~~Printf~~(
  c = '\n';
  result = iscntrl(c);
  printf ("when %c is passed to ~~in~~ iscntru l() = %d",
          c, result);
  getch();
  }
```

output

```
When Q is passed to iscntrl () = 0.
when
    is passed to iscntrl() = 32
```

- toupper

```
#include <stdio.h>
#include <conio.h>

void main(){
 char ch;
 clrscr();
 // Letter to convert to uppercase
 ch='b';

Printf ("%c in uppercase is represented as %c",
        ch,
      getch ();
}
```

Output

```
b in uppercase is re~~pp~~resented as B
```

- tolower()

```
#include <stdio.h>
# include < ctype.h>
# include< conio.h>

Void main(){

  char c, result;
  clrscr(),

   c ='M';
 result = tolower(c);
 Printf("tolower (%c) = %c\n", c, result);
c='m'
result= tolower (c);
printf("tolower (%c) =%c\n", c,result );

c='+';
result = tolower(c);
printf("tolower (%c) = %c\n" , c,result );

getch();
}
```

```
output
tolower(M)=m
tolower(m)=m
tolower(+)=+
```

- logic develop

```
#include<stdio.h>
#include< conio.h>

void main(){
 int i,j;
 clrscr();

for (i=1; i<=5; i++)
{
 for (j=1; j<=i; j++)
  {
    if (i==5 &&j==5)
    {printf ("%d", i);
    }
else
 { printf ("%d ,", i.);}
  }
}
getch(); }
```

```
output
1,2,2,3,3,3,4,4,4,4,5,5,5,5,5
```

storage class:-

- Type

(1) Automatic storage class

```
#include <stdio.h>
#include < conio.h>

void main()
{
  auto int i=1;
  {
     int  i=2;
     {
        auto  int  i=3;
        printf ("\n%d", i);
     }
     printf ("%d", i);
  }
  printf ("%d\n", i);

  getch ();
}
```

output

```
3   2   1
```