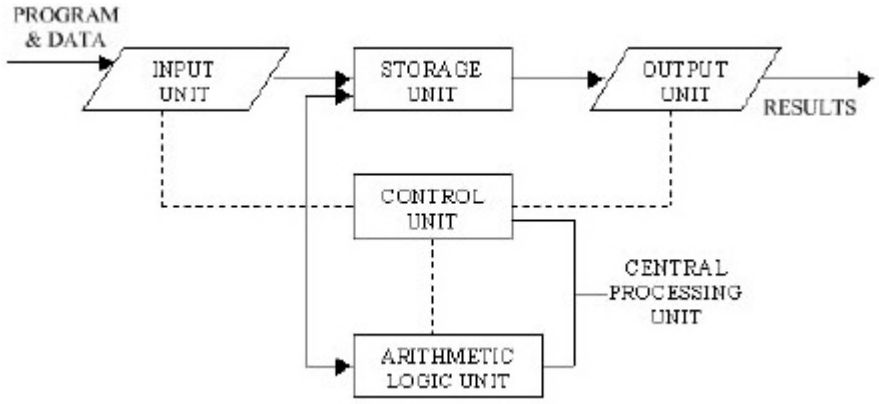
BBDNITM
CSPC Practice Set

1Q. Explain what do you mean by algorithm. What are the properties of good algorithm.
Ans-Algorithm is the process to solve the problem in step by step manner.

properties of good algorithm.:

- Precision – the steps are precisely stated(defined).
- Uniqueness – results of each step are uniquely definedand only depend on the input and the result of the precedingsteps.
- Finiteness – the algorithm stops after a finite number ofinstructions are executed.
- Input – the algorithm receives input.
- Output – the algorithm produces output.
- Generality – the algorithm applies to a set ofinputs.

2Q. Draw the block diagram of digital computer and explain the role of each functional unit.
Ans- Draw the block diagram of digital computer and explain



1. **Input:** This is the process of entering data and programs in to the computer system.

For example key board, mouse, and scanner.

2. **Storage:** The process of saving data and instructions permanently is known as storage. Data has to be fed into the system before the actual processing starts. It is because the processing speed of Central Processing Unit (CPU) is so fast that the data has to be provided to CPU with the same speed.
3. **Processing:** The task of performing operations like arithmetic and logical operations is called processing. The Central Processing Unit (CPU) takes data and instructions from the storage unit and makes all sorts of calculations based on the instructions given and the type of data provided. It is then sent back to the storage unit.

4. **Output:** This is the process of producing results from the data for getting useful information. Similarly the output produced by the computer after processing must also be kept somewhere inside the computer before being given to you in human readable form.

3Q. what are the difference between system software and application software explain with suitable example.

Difference between system software and application software

• System software gets installed when the operating system is installed on the computer while application

software is installed according to the requirements of the user.

• System software includes programs such as compilers, debuggers, drivers, assemblers while application software includes media players, word processors, and spreadsheet programs.

• Generally, users do not interact with system software as it works in the background whereas users interact with application software while doing different activities.

• A computer may not require more than one type of system software while there may be a number of application software programs installed on the computer at the same time.

• System software can run independently of the application software while application software cannot run without the presence of the system software.

4Q What is an operating system write different types of properties of operating system.

Ans. The roles of an operating system vary with the hardware and user programs it was designed to work with. The earliest operating systems were designed to help applications interact with hardware. Operating System acts as resource manager.

Function of OS-1) Memory management 2) Disk Management 3) resource Management 4)I/O management 5) Network Management 6)Security Management 7)File Management.

5Q. What is an operator. Discuss the various types of operator are used in C programming language with example.

Ans- C contains the following operator groups.

- Arithmetic +,-,*,/,%
- Assignment =
- Logical/relational || && !
- Bitwise ^, >>,<<,|, &

6Q. What is storage class, Discuss the various storage classes in C with suitable example.

Storage classes

In C language, each variable has a **storage class** which decides scope, visibility and lifetime of that variable. The following storage classes are most oftenly used in C programming,

1. **Automatic Storage class**
2. **External Storage class**
3. **Static Storage class**
4. **Register Storage class**

Automatic Storage class

A variable declared inside a function without any storage class specification, is by default an **automatic variable**. They are created when a function is called and are destroyed **automatically** when the function exits. Automatic variables can also be called local variables because they are local to a function. By default they are assigned **garbage value** by the compiler.

```
void main()
{
 int detail;
 or
 auto int detail;    //Both are same
}
```

External or Global Storage class

A variable that is declared outside any function is a **Global variable**. **Global** variables remain available throughout the entire program. One important thing to remember about global variable is that their values can be changed by any function in the program.

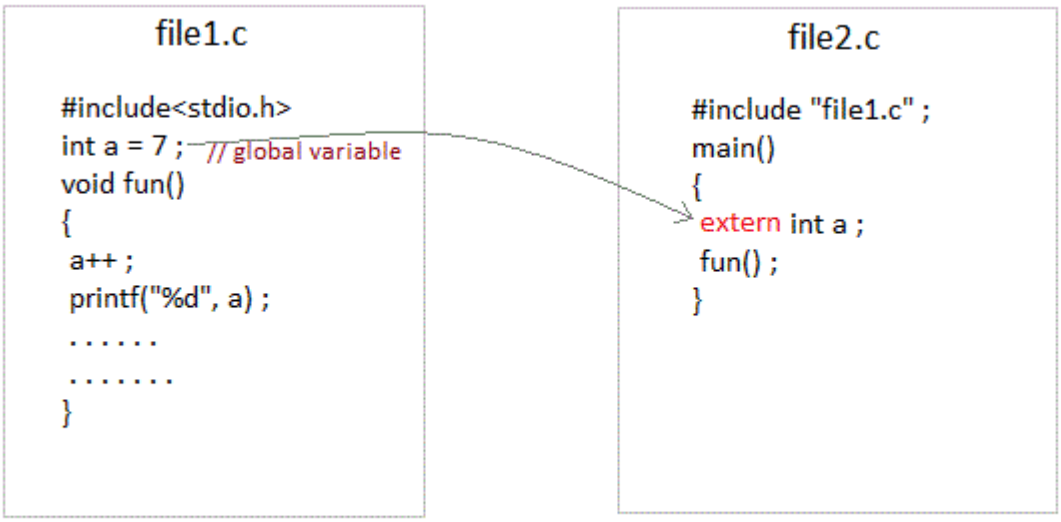
```
int number;
void main()
{
 number=10;
}
fun1()
{
 number=20;
}
fun2()
{
 number=30;
}
```

Here the global variable **number** is available to all three functions.

---

extern keyword

The **extern** keyword is used before a variable to inform the compiler that this variable is declared somewhere else. The **extern** declaration does not allocate storage for variables.



global variable from one file can be used in other using **extern** keyword.

Problem when extern is not used

```
main()
{
  a = 10;       //Error:cannot find variable a
  printf("%d",a);
}
```

Example Using extern in same file

```
main()
{
  extern int x;  //Tells compiler that it is defined somewhere else
  x = 10;
  printf("%d",x);
}

int x;    //Global variable x
```

Static Storage class

A **static** variable tells the compiler to persist the variable until the end of program. Instead of creating and destroying a variable every time when it comes into and goes out of scope, **static** is initialized only once and remains into existence till the end of program. A static variable can either be internal or external depending upon the place of declaration. Scope of **internal static** variable remains inside the function in which it is defined. **External static** variables remain restricted to scope of file in each they are declared.

They are assigned **0 (zero)** as default value by the compiler.

```
void test();   //Function declaration (discussed in next topic)

main()
{
 test();
 test();
 test();
}
void test()
{
 static int a = 0;        //Static variable
 a = a+1;
 printf("%d\t",a);
}
```

output :

```
1       2       3
```

Register Storage class

Register variable inform the compiler to store the variable in register instead of memory. **Register** variable has faster access than normal variable. Frequently used variables are kept in register. Only few variables can be placed inside register.

NOTE : We can never get the address of such variables.

Syntax :

```
register int number;
```

7Q what is the meaning of function prototype, give the classification of function prototype in C.

A function prototype is a function declaration that specifies the data types of its arguments in the parameter list. The compiler uses the information in a function prototype to ensure that the corresponding function definition and all corresponding function declarations and calls within the scope of the prototype contain the correct number of arguments or parameters, and that each argument or parameter is of the correct data type.

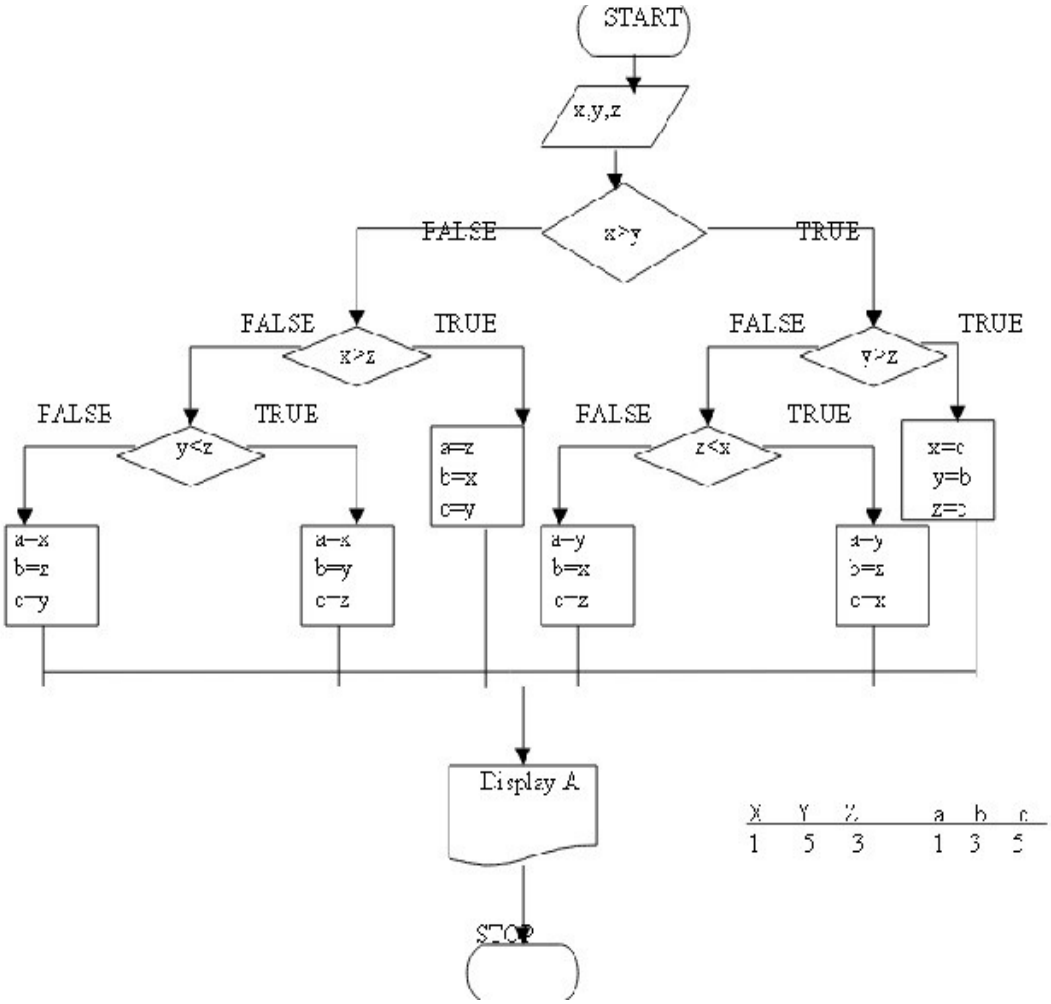
function-prototype-declaration:

```
return_type function_name( argument);
```

classification of function prototype in C

i) Function with no argument and no return type.
ii) Function with no argument and return type.
iii) Function with argument and no return type.
iv) Function with argument and return type.

10Q. Draw the flow chart diagram for arranging three numbers a,b,c in ascending order.



8Q. Write a program using recursive function to calculate the factorial of a positive number.

A recursive function call its self again and again.

```
#include<stdio.h>
int factorial(int n);
int main()
{
   int n;
   printf("Enter an positive integer: ");
   scanf("%d",&n);
   printf("Factorial of %d = %ld", n, factorial(n));
   return 0;
}
int factorial(int n)
{
   if(n!=1)
    return n*factorial(n-1);
}
```

9Q. Print the pattern

1

2 3
4 5 6
7 8 9 10

Ans
```
#include<stdio.h>
int main()
{
   int rows,i,j,k=0;
   printf("Enter number of rows: ");
   scanf("%d",&rows);
   for(i=1;i<=rows;i++)
   {
      for(j=1;j<=i;++j)
        printf("%d ",k+j);
      ++k;
      printf("\n");
   }
}
```

10Q. Write a program in C to find prime number between two interval .

Ans:
```
#include <stdio.h>
int main()
{
  int n1, n2, i, j, flag;
  printf("Enter two numbers(intevals): ");
  scanf("%d %d", &n1, &n2);
  printf("Prime numbers between %d and %d are: ", n1, n2);
  for(i=n1+1; i<n2; ++i)
  {
     flag=0;
     for(j=2; j<=i/2; ++j)
     {
       if(i%j==0)
       {
         flag=1;
         break;
       }
     }
     if(flag==0)
       printf("%d ",i);
```

```
  }
 return 0;
}
```

11Q. Write a program to find the sum of give series
Sum=1!+2!+3!+4!.....N!

C provides for a special type of function called Recursive function. Previously we have seen that in C any function can call any other function. The limiting case is can a function call itself? The answer is yes. A function can call itself. If a function repeatedly keeps calling itself (until certain conditions are satisfied called terminating conditions), then such a function is called a recursive function.

/* Source code to find factorial of a number. */

```
#include<stdio.h>
int factorial(int n);
int main()
{
   int n;
   printf("Enter an positive integer: ");
   scanf("%d",&n);
   printf("Factorial of %d = %ld", n, factorial(n));
   return 0;
}
int factorial(int n)
{
   if(n!=1)
    return n*factorial(n-1);
}
```

12Q. Draw the flow chart diagram and write a program in C to calculate the Fibonacci series .

PROGRAM TO PRINT FIBONACCI SERIES:

```
0 1 1 2 3 5 8 13

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,c,i;

clrscr();
```

```
a=0;
b=1;
printf("\nFIBONACCI SERIES IS \n\n");
printf("%d    %d    ",a,b);
for(i=3;i<=8;i++)
{
     c=a+b;
     printf("%d    ",c);
     a=b;
     b=c;
}
getch();
}
```

Note:Draw the chart on the flow diagram.

13Q. What is the use of break and continue statements. Give suitable example for both statements.

Difference Between `break` and `continue`

break	continue
A `break` can appear in both `switch` and loop (`for`, `while`, `do`) statements.	A `continue` can appear only in loop (`for`, `while`, `do`) statements.
A `break` causes the `switch` or loop statements to terminate the moment it is executed. Loop or `switch` ends abruptly when break is encountered.	A `continue` doesn't terminate the loop, it causes the loop to go to the next iteration. All iterations of the loop are executed even if `continue` is encountered. The `continue` statement is used to skip statements in the loop that appear after the `continue`.
The `break` statement can be used in both `switch` and loop statements.	The `continue` statement can appear only in loops. You will get an error if this appears in switch statement.
When a `break` statement is encountered, it terminates the block and gets the control out of the	When a `continue` statement is encountered, it gets the control to the next iteration of the loop.

`switch` or loop.

A `break` causes the innermost enclosing loop or `switch` to be exited immediately.

A `continue` inside a loop nested within a `switch` causes the next loop iteration.

14Q. What is differentiate between i) array and structure Structure ii) Structure and Union.

i).Array and structure.

Ans-Array is a collection of similar elements. Array elements always stored in contiguous memory location.The declaration of a (fixed-size) array has the following syntax:
int array[100];
which defines an array named array to hold 100 values of the primitive type int.
A Structure is a heterogeneous data type that can store all type of data types.As you may know, we can declare the form of a block of data containing different data types by means of a structure declaration.

ii. Structure and Union- A Structure is a heterogeneous data type that can store all type of data types.As you may know, we can declare the form of a block of data containing different data types by means of a structure declaration.

A `union`, is a collection of variables of different types, just like a structure. However, with unions, you can only store information in one field at any one time. You can picture a `union` as like a chunk of memory that is used to store variables of different types. Once a new value is assigned to a field, the existing data is wiped over with the new data.

15Q. Write a program that read a string from the key board and determine whether the string is a palindrome or not.

c program for palindrome

C program for palindrome or palindrome in c programming: palindrome program in c language, c code to check if a string is a palindrome or not and for palindrome number. This program works as follows :- at first we copy the entered string into a new string, and then we reverse the new string and then compares it with original string. If both of them have same sequence of characters i.e. they are identical then the entered string is a palindrome otherwise not. To perform copy, reverse and compare operations we use strcpy, strrev and strcmp functions of string.h respectively, if you do not wish to use these functions see c programming code for palindrome without using string functions. Some palindrome strings examples are "dad", "radar", "madam" etc.

C program for palindrome

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

Void main()
{
  char a[100], b[100];

  printf("Enter the string to check if it is a palindrome\n");
  gets(a);
```

```
  strcpy(b,a);
  strrev(b);

  if( strcmp(a,b) == 0 )
    printf("Entered string is a palindrome.\n");
  else
    printf("Entered string is not a palindrome.\n");

  return 0;
}
```

Output of program:

```
Enter the string to check if it is a palindrome
able was i ere i saw elba
Entered string is a palindrome.
```

or

Palindrome number in c

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

void main()
{
  int n, reverse = 0, temp;

  printf("Enter a number to check if it is a palindrome or not\n");
  scanf("%d",&n);

  temp = n;

  while( temp != 0 )
  {
    reverse = reverse * 10;
    reverse = reverse + temp%10;
    temp = temp/10;
  }

  if ( n == reverse )
    printf("%d is a palindrome number.\n", n);
  else
    printf("%d is not a palindrome number.\n", n);

  getch();
}
```

18Q. Write a program in C that accept student_ roll, student_ name, student_ marks of physics, chemistry, and mathematics. Print the Student_ roll and Student_ name of the top ten students in order of merit the merit is based on the sum of marks obtain in three subjects.

Ans: Enter students info. Within loop and sum the marks of all students within loop then apply sorting .

16Q.Write a program in C to accept elements of Matrix of size 3x3 and print transpose of it.

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

void main()
{
  int m, n, c, d, matrix[10][10], transpose[10][10];

  printf("Enter the number of rows and columns of matrix ");
  scanf("%d%d",&m,&n);
  printf("Enter the elements of matrix \n");

  for( c = 0 ; c < m ; c++ )
  {
    for( d = 0 ; d < n ; d++ )
    {
      scanf("%d",&matrix[c][d]);
    }
  }

  for( c = 0 ; c < m ; c++ )
  {
    for( d = 0 ; d < n ; d++ )
    {
      transpose[d][c] = matrix[c][d];
    }
  }

  printf("Transpose of entered matrix :-\n");

  for( c = 0 ; c < n ; c++ )
  {
    for( d = 0 ; d < m ; d++ )
    {
      printf("%d\t",transpose[c][d]);
    }
    printf("\n");
  }

  getch();
}
```

```
E:\programmingsimplified.com\java>javac TransposeAMatrix.java

E:\programmingsimplified.com\java>java TransposeAMatrix
Enter the number of rows and columns of matrix
4
2
Enter the elements of matrix
1       2
3       4
5       6
7       8
Transpose of entered matrix:-
1       3       5       7
2       4       6       8
```

22Q. Explain the flowing string function in C
Strrev(), strlen(), stecpy(),strcmp(),strcat()

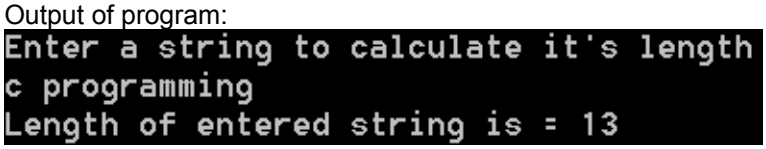
String length strlen()

This program prints length of string, for example consider the string "c programming" it's length is 13. Null character is not counted when calculating string length. To find string length we use strlen function of string.h.

C program to find string length

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

Void main()
{
  char a[100];
  int length;

  printf("Enter a string to calculate it's length\n");
  gets(a);

  length = strlen(a);

  printf("Length of entered string is = %d\n",length);

  getch();
}
```

Output of program:



c program to compare two strings strcmp()

This c program compares two strings using strcmp, without strcmp and using pointers. For comparing strings without using library function see another code below.

C program to compare two strings using strcmp

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

void main()

{
  char a[100], b[100];

  printf("Enter the first string\n");
  gets(a);

  printf("Enter the second string\n");
  gets(b);

  if( strcmp(a,b) == 0 )
    printf("Entered strings are equal.\n");
  else
    printf("Entered strings are not equal.\n");

  getch();
}
```

string copying in c programming,stecpy()

This program copy string using library function strcpy, to copy string without using strcpy see source code below in which we have made our own function to copy string.

C program to copy a string

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

void main()

{
  char source[] = "C program";
  char destination[50];

  strcpy(destination, source);

  printf("Source string: %s\n", source);
```

```
   printf("Destination string: %s\n", destination);

  getch();
}
```

c program to concatenate strings,strcat()

This program concatenates strings, for example if the first string is "c " and second string is "program" then on concatenating these two strings we get the string "c program". To concatenate two strings we use strcat function of string.h, to concatenate without using library function see another code below which uses pointers.

C code

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

Void main()
{
   char a[100], b[100];

   printf("Enter the first string\n");
   gets(a);

   printf("Enter the second string\n");
   gets(b);

   strcat(a,b);

   printf("String obtained on concatenation is %s\n",a);

   getch();

}
```

Output:

```
Enter the first string
under
Enter the second string
stand
String obtained on concatenation is understand
```

Reverse string ,strlen()

This program reverses a string entered by the user. For example if a user enters a string "reverse me" then on reversing the string will be "em esrever". We show you three different methods to reverse string the first one uses strrev library function of string.h header file and in second we make our own function to reverse string using pointers,reverse string using recursion and Reverse words in string. If you are using first method then you must include string.h in your program.

C programming code

```
/* String reverse in c*/

#include<stdio.h>
#include<string.h>

Void main()
{
  char arr[100];

  printf("Enter a string to reverse\n");
  gets(arr);

  strrev(arr);

  printf("Reverse of entered string is \n%s\n",arr);

  getch();
}
```

```
Enter a string to reverse
hello
Reverse of entered string is
olleh
```

17Q. What is pointer Variable? Why they are required? Write a program in C to read two integer X and Y and Swap the contents of the Two variable X and Y using pointer.

or

Difference between call by value and call by reference

Ans- A pointer variable always keep the address of other variable.

By call by value

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

void swap( int , int  )
  // call by refrence
{
   int t ;
   t = x ;
   x = y ;
   y = t ;
   printf( "\nx = %d y = %d", *x,*y);
}

int main( )
```

```
{
   int a = 10, b = 20 ;
   swap ( a, b ) ;
  // passing the address of values to be swapped
   printf ( "\na = %d b = %d", a, b ) ;
   getch();
}
```

By call by reference

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

void swap( int , int  )
  // call by refrence
{
   int t ;
   t = *x ;
   *x = *y ;
   *y = t ;
   printf( "\nx = %d y = %d", *x,*y);
}

int main( )
{
   int a = 10, b = 20 ;
   swap ( &a, &b ) ;  // passing the address of values to be swapped
   printf ( "\na = %d b = %d", a, b ) ;
   getch();
}
```

18Q.What do you mean by C pre-processor Directive? Write its all types with suitable example.

Preprocessor extends the power of C programming language. Line that begin with # are called preprocessing directives. we introduced the `define` compiler directive which defines symbolic names for strings of characters. Such a string of characters can be arbitrary, for example a sequence of characters representing a numeric constant. These names can then be used anywhere in the program instead of the string itself. The C preprocessor replaces these symbolic names with the specified strings prior to compiling the program. We have seen examples where using names for arbitrary strings makes it easy to change all occurrences of these names by merely changing the definitions. It also makes for easier reading and debugging of programs by allowing the programmer to use a name which has some meaning rather than some ``magic

number". The definition is called a **macro** and the preprocessor performs a **macro expansion** when it substitutes the string for the name. A macro definition takes the form:

#define <symbol_name> <substitution_string>

Type of C pre-processor Directive

- **Macros**
- **String Replacement**
- **File inclusion**
- **Conditional Compilation**

Note: Go through Class notes

19Q What do you mean by sorting ? Write a program in C to sort the given n positive integer in ascending and descending order.

Program on Sorting element in descending order.

```
#include<stdio.h>

#include<conio.h>

void main()

{

int a[10],i,j,temp;

printf("enter the element");

for(i=0;i<10;i++)

{

scanf("%d",&a[i]);

}

for(i=0;i<10;i++)

{

for(j=i+1;j<10;j++)

{

if(a[i]<a[j])

{
```

```
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
printf("elements in desending order\n");
for(i=0;i<10;i++)
{
printf("%d",a[i]);
}
getch();
}
```

Program on Sorting element in Ascending order.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10],i,j,temp;
printf("enter the element");
for(i=0;i<10;i++)
{
scanf("%d",&a[i]);
}
for(i=0;i<10;i++)
{
for(j=i+1;j<10;j++)
{
```

```
if(a[i]>a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
printf("elements in asending order\n");
for(i=0;i<10;i++)
{
printf("%d",a[i]);
}
getch();
}
```

20Q Write a program in C to store the floating point number(float number) in two matrix A and B of size 4x4 each further print the summation and multiplication of two matrix and store the summation and multiplication in matrix C and D.

Ans: summation of matrix

```
#include<stdio.h>
main()
{
        int a[5][5],b[5][5],c[5][5];
        int i,j,m,n;
        /*GET THE MATRIX FROM THE USER*/
        printf("enter matrix A:\n");
        for(i=0;i<m;i++)
        for(j=0;j<n;j++)
                scanf("%d",&a[i][j];
        printf("enter matrix B:\n");
        for(i=0;i<m;i++)
        for(j=0;j<n;j++)
                scanf("%d",&b[i][j];
        /*ADD THE MATRICES*/
        for(i=0;i<m;i++)
```

```
        for(j=0;j<n;j++)
                c[i][j]=a[i][j] + b[i][j];
        /*TO PRINT THE SUM*/
        printf("\n THE SUM OF MATRICES:\n");
        for(i=0;i<m;i++)
            {
               for(j=0;j<n;j++)

                 printf("%d\t",c[i][j];
                 printf("/n/");
            }
}
```

OUTPUT:

```
Enter matrix A:
1       1       1
1       1       1
1       1       1
Enter matrix B:
2       2       2
2       2       2
2       2       2


THE SUM OF MATRICES:
3       3       3
3       3       3
3       3       3
```

PRODUCT OF MATRICES

```
      #include<stdio.h>
       main()
        {
         int a[5][5],b[5][5],c[5][5];
         int i,j,k,m,n,p,q;
         printf("\n ENTER ORDER OF MATRIX A:");
         scanf(%d%d",&m,&n);
         printf("\n ENTER ORDER OF MATRIX B:");
         scanf(%d%d",&p,&q);
         if (n!=p)
         {
                 printf("\n\tCANNOT MULTIPLY");
                 exit();
         }
```

```
        /*GET THE MATRIX FROM THE USER*/
        printf("enter matrix A:\n");
        for(i=0;i<m;i++)
        for(j=0;j<n;j++)
                scanf("%d",&a[i][j];
        printf("enter matrix B:\n");
        for(j=0;j<p;j++)
        for(k=0;k<q;k++)
                scanf("%d",&b[j][k];
        /*PRODUCT OF MATRICES*/
        for(i=0;i<m;i++)
        {
                for(k=0;k<q;k++)
                    {
                        c[i][k]=0;
                        for(j=0;j<n;j++)
                        c[i][k]+=a[i][j] - b[j][k];
                }
        }
        /*TO PRINT THE PRODUCT OF MATRICES*/
        printf("\n THE PRODUCT OF MATRICES:\n");
        for(i=0;i<m;i++)
            {
               for(k=0;k<q;k++)
                 printf("%d\t",c[i][k];
                 printf("\n");
            }
}
```

OUTPUT:

```
Enter matrix A:
2       2       2
2       2       2


Enter matrix B:
3        3
3        3
3       3
THE PRODUCT OF MATRICES:
18       18
18       18
```

21Q What is difference Between

i) While and DO- While loop

Ans. The general form is :

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

where statement is either an empty statement, a single statement, or a block of statements. The condition may be any expression, and true is any nonzero value. The loop iterates while the condition is true.When the condition becomes false, program control passes to the line of code immediately following the loop.

Unlike for and while loops, which test the loop condition at the top of the loop, the do-while loop checks its condition at the bottom of the loop. This means that a do-while loop always executes at least once. The general form of the do-while loop is:

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

ii) Flow chart and Algorithm: Flow Chart is the pictorial representation of a program and algorithm is step by step process approach to solve the program.

22Q. Write a C program to calculate the Sum of the Series 1/1! + 2/2! + 3/3! + ……N/N!

```
#include <stdio.h>

double sumseries(double);

void main()

{

double number,sum;

printf("\n Enter the value:  ");

scanf("%lf", &number);

sum = sumseries(number);

printf("\n Sum of the above series = %lf ", sum);

}
```

```
double sumseries(double m)
{
 double sum2 = 0, f = 1, i;
 for (i = 1; i <= m; i++)
 {
 f = f * i;
 sum2 = sum2 +(i / f);
}
return(sum2);
}
```

23Q. a)Convert $B2D_{16}$ to decimal.

Using the same method to convert hexadecimal to decimal.

$= (Bx16^2)+(2x16^1)+(Dx16^0)$

$= (11x16^2)+(2x16^1)+(13x16^0)$

$= 2816 + 32 +13$

$= 2861_{10}$

Therefore $B2D_{16} = 2861_{10}$.

Substract 001100 from 0011010 and find the 2's complement of result?

0011010- 001100= 00001110 1's compliment=11110001

2's compliment=11110001

+1

11110010

(b) Explain the role of translator, linker and loader in C programming language?

Ans- Translator –Translate high level language into low level language ex. Compiler , assembler.

Linker-Bind the object code which is widely available in separate different memory allocation.

Loader- Load the linked object code in memory.

24Q Write a program to create an empty file and input a string in this file and copy the content of this file into another file.

```
#include<stdio.h>
int main(){
 FILE *p,*q;
 char file1[20],file2[20];
 char ch;
 printf("\nEnter the source file name to be copied:");
 gets(file1);
 p=fopen(file1,"r");
 if(p==NULL){
 printf("cannot open %s",file1);
 exit(0);
 }
 printf("\nEnter the destination file name:");
 gets(file2);
 q=fopen(file2,"w");
 if(q==NULL){
printf("cannot open %s",file2);
exit(0);
 }
 while((ch=getc(p))!=EOF)
 putc(ch,q);
 printf("\nCOMPLETED");
 fclose(p);
 fclose(q);
return 0;
}
```