Programming

Passing an array to a function

Declaration: Int sum_array(int [], int);

Call: int s= sum_array(a,n);

Definition: int sum_array(int x[], int m)

```
                {
                        int sa=0;
                        for(int i=0;i<m;i++)
                        {
                                sa=sa+x[i];
                        }
                        return(sa);
                }
```

Passing a 2D array to a function

//Prepare a C++ program to read a m x n matrix and find the average of each column. Read the matrix using the function named as matread, find the average of the column elements using the function colavg.

```
#include<iostream.h>

void main()
{
    int m,n;
    float a[10][10];
    void matread(int p, int q, float b[ ][10]);
    void colavg(int p, int k, float b[ ][10]);
    cout <<"\nEnter the size of the matrix.\n";
    cin >>m >>n;
    cout <<"\nEnter the elements of the matrix.\n";
    matread(m, n, a);
    cout <<"\n The average of each column:\n";
    for(int j = 0; j < n; j++)
          colavg(m, j, a);
}
```

```
void matread(int p, int q, float b[ ][10])
{
    for(int i = 0; i < p; i++)
        for(int j = 0; j < q; j++)
           cin >>b[i][j];
}
void colavg(int p, int k, float b[ ][10])
{
    float sum = 0;
    for(int i = 0; i < p; i++)
           sum += b[i][k];
    float cavg = sum / p;
     cout <<" The average of column " <<k + 1 <<" is " <<cavg
    <<".\n";
}
```

Structures and Functions:

1) Members as arguments
2) Entire structure as argument
3) Address of structure as argument

Members as arguments:

```
Void functn (int empno, char name[], float salary);

functn(e.empno, e.name, e.salary);

Void functn (int empno, char name[], float salary)
{
        Statements;
}
```

Entire structure as argument:

```
Void  functn(struct emp e);

Functn(e);

Void  functn(struct emp e)
{
    statement;
}
```

Address of structure as argument

```
Void functn(struct emp *);

Functn(&e);

Void functn(struct emp *e)
{

Statements;

}
```

Accessing members inside the function

```
e->empno,
 e->empname,
e->salary
```