

Note: There are three ways to pass a 2D array to a function:

1. The parameter is a 2D array

```
int array[10][10];
void passFunc(int a[][10])
{
    // ...
}
passFunc(array);
```

2. The parameter is an array containing pointers

```
int *array[10];
for(int i = 0; i < 10; i++)
    array[i] = new int[10];
void passFunc(int *a[10]) //Array containing pointers
{
    // ...
}
passFunc(array);
```

3. The parameter is a pointer to a pointer

```
int **array;
array = new int *[10];
for(int i = 0; i < 10; i++)
    array[i] = new int[10];
void passFunc(int **a)
{
    // ...
}
passFunc(array);
```

1. Write a program to add two numbers using pointers.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int *p1,*p2,sum;
    cout << "enter two no's: " << *p1 << *p2;
    cout << "The sum of the two numbers is: " << *p1+*p2;
    getch();
}
```

2. Write a program to find the maximum number in array using pointer.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int *max, i, a[5];
    cout<< "enter element for the array: ";
    for(i=0;i<5;i++)
        cin>>*(a+i);
    max=a;
    for(i=1;i<5;i++)
    {
        if(*max<*(a+i))
            max=a+i;
    }
    cout<<"maximum no= " <<*max;
    getch();
}
```

3. Write a program to create an integer array. Ask the user to specify the length of the array, then read the array and find the sum & the average array cells.

```
#include <iostream.h>
void main( )
{
    int d, i, sum, avg;
    sum=0;
    cout << "How long the size of the array: " ;
    cin >> d;
    int *a=new int [d];
    cout << "enter the array" << endl;
    for ( i=0;i<d; i++)
    {
        cin >> *(a+i);
        sum=sum+a[i];
    }
    avg=sum/d;
    cout << "sum= " << sum << endl;
    cout << "avg= " << avg;
}
```

4. Write a program to create a two-dimensional array. The dimensions of the array are specified by the user when running the code. Find the sum of the first row.

```
#include <iostream.h>
#include<conio.h>
void main ()
{
    int d1,d2,sum,i,j;
    sum=0;
    cout << "enter the dimension of the array";
    cin >> d1 >> d2;
    int **a=new int *[d1];
    for (i=0; i< d1; i++)
        a[i]=new int[d2];
    for (i=0 ; i< d1 ; i++)
        for (j=0; j< d2 ; j++)
            cin >> *((a+i)+j);
    for (i=0 ; i< d1 ; i++)
        for (j=0; j< d2 ; j++)
            if (i==0)
                sum=sum+ *((a+i)+j);
    cout << "The sum of the first row is:" << sum<<endl;
}
```

5. write a program to print a two-dimensional array (5x5) using two different functions. The first function prints the array without using the pointer and the second function prints the array by using the pointer.

```
#include <iostream.h>
void displaywithoutpointer(int array[][5] )
{
    for (int i=0;i<5;i++)
    for (int j=0;j<5;j++)
    cout<<"array=["<<i<<"["<<j<<"]="<< array[i][j] <<"\n";
}
void displaywithpointe(int *array )           // pass the array as pointer
{
    int i=0;
    while ( i<25)                             //to display 25 element
    {cout<<"array=["<<i<<"]="<<*array <<"\n";
    *array++;                                  //increment the location of
    pointer
    i++;}}
main()
{
    int array[5][5];
    cout << "Enter Here the Array\n";
    for (int i=0;i<5;i++)
    for (int j=0;j<5;j++)
    cin>>array[i][j];
    displaywithoutpointer(array);
    displaywithpointe(array);                  //sent location first element
}
```

6. Write a function that uses pointers to copy an array of double.

```
#include<iostream>
double* copy(double a[], int n)
{
    double* p = new double[n];
    for (int i = 0; i < n; i++)
        p[i] = a[i];
    return p;
}
void print(double c[],int d)
{for (int i=0;i<d;i++)
cout << c[i] <<"\t";
}
int main()
{
    double a[8] = {22.2,
33.3,44.4,55.5,66.6,77.7,88.8,99.9};
    print(a,8);
    double* b = copy(a,8);
    a[2] = a[4] = 11.1;
    print(a,8);
    print(b,8);
}
```

7. Write a function that uses pointers to search for the address of a given integer in a given array. If the given integer is found, the function returns its address; otherwise it returns NULL.

```
#include <iostream.h>
int* location(int a[],int n, int target)
{ for (int i = 0; i < n; i++)
if (a[i] == target) return &a[i];
return NULL;
}
int main()
{ int a[8] = {22,33, 44,55,66,77,88,99},*
p, n;
do
{ cin >> n;
if (p = location(a,8,n)) cout << p << ", " <<
*p << endl;
else cout << n << " was not found.\n";
} while (n > 0);
}
```

8. Write a function that is passed an array of n pointers to floats and returns a newly created array that contains those n float values.

```
#include <iostream.h>
float* duplicate(float* p[],int n)
{ float* const b = new float[n];
  for (int i = 0; i < n; i++)
    b[i] = *p[i];
  return b;
}
void print(float [],int);
void print(float* [],int);
int main()
{ float a[8] = {44.4, 77.7,22.2,88.8,66.6,33.3,99.9,55.5};
  print(a,8);
  float* p[8];
  for (int i = 0; i < 8; i++)
    p[i] = &a[i]; // p[i] points to a[i]
  print(p,8);
  float* const b = duplicate(p,8);
  print(b,8);
}
```

9. Define a string then print it using a pointer.

```
#include<iostream>
#include<conio.h>
void main()
{ char title[] = "c++ language",
  i=0;
  while(*(title+i) != '\0')
    Puthcar(*(title+ i++));
  getch();
}
```

10. Write a program to define a pointer to a character string containing the name of the nth month. Then print the name of the month depending on your selection from 1-12.

```
#include<iostream>
#include<conio.h>
void main()
{ int selection
  Char *name[] = {"Illegel month","January", "February", "March", "April",
  "May", "June", "July", "August", "September", "October", "November",
  "December"};
  cout << " Select the month: ";
  cin>> selection;
  selection < 1 || selection > 12 ? cout<< name[0] : cout<< name[selection];
  getch();
}
```

11. Write a program to add a constant to a two-dimensional array with 4 rows and 5 columns. Print the array element after addition using pointers.

```
#include<iostream>
#include<conio.h>
void main()
{ int table[4][5] =
  {{13,15,17,19,21},{20,22,24,25,26},{31,33,35,3
  7,39},{49,42,44,46,48}};
  int x,i,j;
  cout<< "Please enter the constant: ";
  cin>> x;
  for(i=0;i<4;i++)
  {
  for (j = 0; j<5;j++)
    cout<< *((table+i)+j) +=x;
  cout<< "\n";
  }
}
```

12. Write a function that is passed an array of n pointers to floats and returns a pointer to the maximum of the n floats.

```
#include <iostream.h>
float* max(float* p[],int n)
{
    float* pmax = p[0];
    for (int i = 1; i < n; i++)
        if (*p[i] > *pmax) pmax = p[i];
    return pmax;
}

void main()
{
    float a[8] = {44.4, 77.7,22.2,88.8,66.6,33.3,99.9,55.5};
    float* p[8];
    for (int i = 0; i < 8; i++)
        p[i] = &a[i]; // p[i] points to a[i]
    float* m = max(p,8) ;
    cout << m << " " << *m << endl;
}
```

13. Write a program to read an integer matrix a[5][5], then arrange the rows in decreasing order depending on the maximum number in each row. Then print the arranged matrix on the screen.

```
#include<iostream.h>
#include<conio.h>
int main()
{ int i, j, k, a[5][5], max[5], t;
  cout<<"Please enter the array's elements"<<endl;
  for(i=0;i<5;i++)
    for(j=0;j<5;j++)
      cin>>a[i][j]<<endl;
  for(i=0;i<5;i++)
  {
    *(max+i) = (*(a+i));
    for(j=1;j<5;j++)
      if(*(a+i)+j) < *(max+i))
        *(max+i) = (*(a+i)+j);
  }
  for(j=0;j<5;j++)
    if(*(max+i) < *(max+j))
    {
      t = *(max + i);
      *(max+i) = *(max+j);
      *(max+j) = t;
      for (k=0;k<5;k++)
      {
        t = (*(a+i)+k);
        (*(a+i)+k) = (*(a+j)+k);
        (*(a+j)+k) = t;
      }
    }
  clrscr();
  cout<<"\n\t"<<"the new aarray is: "<<"\n\n";
  for(i=0;i<5;i++)
    for(j=0;j<5;j++)
    {
      cout<<"\t"<< (*(a+i)+j);
      cout<<"\n";
    }
  getch();
}
```

14. Write the following function that is passed an array of n pointers to floats and returns a newly created array that contains those n float values in reverse order.

```
#include <iostream.h>
float* mirror (float* p[],int n)
{
    float* pnew = new float [n];
    for (int i = 0; i < n; i++)
        *pnew[i] = *p[n-i-1];
    return pnew;
}

void main()
{
    float a[8] = {44.4, 77.7,22.2,88.8,66.6,33.3,99.9,55.5};
    float* m = mirror (a,8) ;
    for (int i=0; i<8;i++)
        cout << *(m+i) << endl;
}
```

- Q1. Using a pointer. Write a program to read a series of letter grades terminated by a dollar sign, as in
 A A F B C F D B A C C B D F \$
 The program should count and print the number of A's, B's, C's, and so on.
- Q2. A magic square is a square array of integers such that every row, column, and diagonal has the same sum, as in
 2 7 6
 9 5 1
 4 3 8
- Using a pointer of array, write a program to check whether a square is magic by computing and comparing the sums of its rows, columns, and diagonals.
- Q3. Using a pointer of array. Write a program to read an integer array a[10], then find the two largest values among the 10 numbers.
- Q4. Using a pointer of array. Write a program that determine if an one dimensional array of ten integer numbers is stored in increasing order, in decreasing order, or is unordered. Your program should print 0 if the array is unordered, 1 if sorted in increasing order, and 2 if sorted in decreasing order.
- Q5. Another name of a two-dimensional array is matrix A. matrix is called a square matrix if its number of rows is the same as the number of columns. Using a pointer of array, write a program that determine whether a given square matrix is symmetric or not, and print 1 under its name if it is, and 0 if it is not. A matrix mat is symmetric if $\text{mat}[i][j] = \text{mat}[j][i]$ for all legal values of i and j, except when $i=j$.
- Q6. Using a pointer, write a program that calculates and print the average of several integers. Assume the last value read in the sentinel 9999. A typical input sequence might be
 10 8 11 7 9 9999
 Indicating that the average of all the values preceding 9999 is to calculated.
- Q7. Using a pointer of array. Write a program that reverse a list stored in a on dimensional array, and print the reversed list in the same array. For example, if the list is {1, 2, 3}, you should have {3, 2, 1} after the program is executed.
- Q8. Using a pointer of array, write a program to enter an array of four rows & four columns then do the following:
 a) Find The sum of all rows separately. (When you press 1)
 b) Find the sum of second diagonal elements. (when you press 2)
 c) Find the multiplication of triangular elements under the main diagonal Elements. (when you press 3)
 d) Print the numbers that divided by 4 or 10 without any reminder. (otherwise)
- Q9. Using a pointer of array, write a program to enter an array of integer numbers a[4][4] then :-
 a) Interchange the triangular elements above the second diagonal with triangular elements below the second diagonal.
 b) Add a constant to fourth column.
 c) Add 1 to 1st row, 2 to 2nd row, 3 to 3rd row & 4 to 4th row.
 d) Arrange array elements in increasing order.
 e) Print the maximum number & their location in triangular elements below the main diagonal