```
// Author - Harsh Dixit
// Mail - harsh02.dixit@gmail.com
// Linkedin - https://www.linkedin.com/in/harsh-dixit10


// POINTER TO AN ARRAY --->  pointer to an array is a pointer that points at the first
element of arrary

#include<iostream>
using namespace std;
int main()
{
        int nums[] = {5,6,7};
        int (*ptr)[3] = &nums;  //pointer to array of 3 integers

        cout << (*ptr)[1] <<endl;

        int *ptr2 = nums;

        cout << *(ptr2 +2) << endl;


        /* NOTE: While creating pointer to an array
                 int nums[3] = {1,2,3};
                 int (*ptr)[3] = &nums;
                 the parenthesis around *ptr isneccessary bcz the de-refrence operator *
                 has lower precedance than the array subscript operator[].
        */



        // ARRAY OF POINTERS
        // An array of pointers is an array where each element is a pointer to a memory
location

        int nums[5] = {1,3,5,7,9};
        int *arr[5];

        for (int i=0; i<5; i++)
        {
                arr[i] = &nums[i];
                cout << *arr[i] << endl;
        }
        return 0;
}

//---------------------------------------------------------------------------------------
        // CASE: When Array passed to a function
        #include<iostream>
    using namespace std;
        int solve(int arr[] , int size)  // this is pointer to arr array
        {
                cout << "Size of an array inside the function: " << sizeof(arr) << endl;
        }

        int main()
        {
                int arr[5] = {2,3,6,8,3};
                cout << "Size of an array inside main: " << sizeof(arr) << endl;
                solve(arr , 5);
```

```
       }

       /* NOTE: When array is passed to a function not the whole array is passed
       but actually base address of array is passed to the function. We can also write
       int arr[] in solve functio to int *arr
       */


    // Question 01
    #include<iostream>
    using namespace std;
    int solve(int arr[] , int size)
       {
              cout << arr << endl;
              cout << &arr << endl; // gives base address of pointer array
              cout << "Size of an array inside the function: " << sizeof(arr) << endl;
       }

       int main()
       {
              int arr[5] = {2,3,6,8,3};
              cout << arr << endl; // gives base address of array
              cout << &arr << endl; // gives base address of array
              cout << "Size of an array inside main: " << sizeof(arr) << endl;
              solve(arr , 5);
       }


//-------------------------------------------------------------------------------

// POINTER TO POINTER
    #include<iostream>
    using namespace std;
    int main(){
    int a = 90;
    int *p = &a;
    int **p1 = &p; //double pointer
    int ***p2 = &p1;  //triple pointer
    int ****p3 = &p2;  //multi pointer
    cout << a << endl;
    cout << &a << endl;
    cout << p << endl;  // address of a
    cout << *p << endl;  // value of a
    cout << &p << endl;  // address of p pointer
    cout << p1 << endl;  // address of p
    cout << **p1 << endl; // value of a
    cout << &p1 << endl;  // address of p1 pointer
    cout << p2 << endl;   // address of p1
    cout << ***p2 << endl;  // value of a
    cout << &p2 << endl;    // address of p2 pointer
    cout << p3 << endl;    // address of p2
    cout << ****p3 << endl;   //value of a
    cout << &p3 << endl;    // address of p3 pointer
}
```