1. What happens when an array is passed as a pointer in function.
   1. When an array is passed as a pointer, the function receives the first element of array instead of full array.
2. Print the array from the function.

#include <stdio.h>

// Function that takes an array and its size as parameters

void printArray(int\* arr, int size) {

for (int i = 0; i < size; i++) {

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

int myArray[] = {1, 2, 3, 4, 5};

int size = sizeof(myArray) / sizeof(myArray[0]);

printf("Array in main(): ");

printArray(myArray, size);

return 0;

}

1. What will happen if you try to access a value which is out of array boundary
   1. When accessed outside array boundary it results in undefined output
2. What happens if you try to add an element to an array which falls outside the boundary.
   1. When you try to add new element it results in undefined output
3. Write a program to get maximum repeating element in an array

#include<stdio.h>

int main()

{

int a[100];

int i,j,k,element;

int maxcount = 0;

int count = 0;

int size;

// take input of the array size

printf("Enter the size of an array: \n");

scanf("%d",&size);

// take all the elements in the array

printf("Enter the elements in an array: \n");

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

{

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

}

// Execute the conditions

// Outer loop to check one by one from each index

for (j = 0; j<size; j++)

{

count = 1; // Why count is declared here not in the second loop because, count is initially considered for each element as 1

// second loop to check and compare next elements;

for ( k = j+1 ; k<size ; k++)

{

// comparing each element from previous loop with the next elements therefore j+1 as the condition

if (a[j]==a[k])

{

// if satisfied then increment the count

count++;

}

// Now, this loop keeps track of all maximum counts of each element, if any element exceeds then replace the current element and store it

if (count>maxcount)

{

maxcount = count;

element = a[j];

}

}

}

printf("%d is the element repeated for %d times \n",element,maxcount);

return 0;

}