| **Practical Number** | 05- II |
| --- | --- |
| **Areas covered** | Single Dimensional Arrays |

Use the following array for the questions below.   
int arr[] = {1, 2, 2, 4, 5, 6, 6};

int uniq[7] = {1, 2, 4, 5, 6, 0, 0 }

1. Write a C Program to find the total number of elements of the given array.
2. Write a C Program to find the index of a given element.
3. Write a C Program to get the count of all duplicate items on the above array.
4. Write a C Program to count the number of Unique Elements on the above array.
5. Declare two single dimensional array with the size given by the user and find , display the followings;

* Scalar Sum ( Adding values of each element of an array)
* Vector Sum (Adding values of each relative elements of an array and store them in third array)
* Vector Product (Multiply values of each relative elements of an array and store them in third array)
* Scalar Product (Multiply values of each relative elements of an array and store them in third array. After placing the values in third array add all the values)

1. Write a C Program to find the total number of elements of the given array.

"sizeof" operator gives the size of the array in bytes so, to get the length of the array we divide the size of the array with the size of the first element.

In this program, we need to count and print the number of elements present in the array.

A number of elements present in the array can be found by calculating the length of the array.

1. #include <stdio.h>
3. **int** main()
4. {
5. //Initialize array
6. **int** arr[] = {1, 2, 3, 4, 5};
8. //Number of elements present in an array can be calculated as follows
9. **int** length = **sizeof**(arr)/**sizeof**(arr[0]);
11. printf("Number of elements present in given array: %d", length);
13. **return** 0;
14. }

Write a C Program to find the index of a given element.

#include <stdio.h>

int main() {

int arr[] = {2, 4, 6, 8, 10};

int x = 8;

int arrLen = sizeof arr / sizeof arr[0];

int index = -1;

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

if (arr[i] == x) {

index = i;

break;

}

}

if (index > -1) {

printf("Index : %d\n", index);

} else {

printf("%d is not present in this array.\n", x);

}

return 0;

}

Write a C Program to get the count of all duplicate items on the above array.

1. #include <stdio.h>
3. **int** main()
4. {
5. //Initialize array
6. **int** arr[] = {1, 2, 3, 4, 2, 7, 8, 8, 3};
8. //Calculate length of array arr
9. **int** length = **sizeof**(arr)/**sizeof**(arr[0]);
11. printf("Duplicate elements in given array: \n");
12. //Searches for duplicate element
13. **for**(**int** i = 0; i < length; i++) {
14. **for**(**int** j = i + 1; j < length; j++) {
15. **if**(arr[i] == arr[j])
16. printf("%d\n", arr[j]);
17. }
18. }
19. **return** 0;
20. }
21. Write a C Program to count the number of Unique Elements on the above array.

#include <stdio.h>

int countDistinct(int a[], int n) //Function Definition

{

int i, j, count = 1;

//Traverse the array

for (i = 1; i < n; i++) //hold an array element

{

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

{

if (a[i] == a[j]) //Check for duplicate elements

{

break; //If duplicate elements found then break

}

}

if (i == j)

{

count++; //increment the number of distinct elements

}

}

return count; //Return the number of distinct elements

}

int main()

{

int n; //Declare array size

printf("Enter the number of elements \n");

scanf("%d",&n); //Initialize the array size

int a[n]; //Array Declaration

printf("Enter the array elements : ");

for (int i=0;i<n;i++)

scanf("%d",&a[i]); //Initialize the array elements

int c= countDistinct(a,n); //Function Call

printf("The number of distinct elements are %d",c); //Print the number of distinct elements

return 0;

}

Copy

#include <stdio.h>

#include <stdlib.h>

int countDistinct(int a[], int n) //Function Definition

{

int i, j, count = 1;

//Traverse the array

for (i = 1; i < n; i++) //hold an array element

{

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

{

if (a[i] == a[j]) //Check for duplicate elements

{

break; //If duplicate elements found then break

}

}

if (i == j)

{

count++; //increment the number of distinct elements

}

}

return count; //Return the number of distinct elements

}

int main()

{

int n=4, a[4]={1,2,3,3};

int c= countDistinct(a,n); //Function Call

printf("The number of distinct elements are %d",c); //Print the number of distinct elements

return 0;

}