

Lab4

Bibek Dhungana

CODE:

Method 1:

```
/*for Visual studio 2019 to use scanf with our error*/
#define _CRT_SECURE_NO_WARNINGS
/*
*
* AUTHOR: Bibek Dhungana
* FILENAME: lab4.c
* DATE: 2/26/2021
* SPECIFICATION: This program prints out the unique numbers in a array.
* FOR: CS 1412 Programming Principle 2 Section 504
*/

/*Importing all the required library*/
#include <stdio.h>
#include <stdlib.h>

/*Function prototype for uniqueElement array*/
int* uniqueElementArray(int* myArray, int length, int* finalArrayLength);

int main(void) {

    /*Initializing all the required variables*/
    int arrayLength;
    int uniqueArrayLength;

    /*Printing the information to the user*/
    printf("Print all unique elements of an array:\n");
    printf("-----\n");

    /*asking the user to enter the length of array*/
    printf("Input the number of elements to be stored in the array:");
    scanf("%d",&arrayLength);

    /*Dynamically allocating the memory for storing element in the array*/
    int* givenArray = (int*)calloc(arrayLength, sizeof(int));

    /*taking input from the user and storing in the array myArray*/
    printf("Input %d elements in the array:\n", arrayLength);
    for (int i = 0; i < arrayLength; i++) {
        printf("element-%d:", i);
        scanf("%d",&givenArray[i]);
    }
}
```

```

}

/*calling unique element function to get unique element array from given array*/
int* finalArray = uniqueElementArray(givenArray, arrayLength,&uniqueArrayLength);

/*Printing out the unique element of the array*/
printf("The unique elements found in the array are:\n");
for (int i = 0; i < uniqueArrayLength; i++) {
    printf("%d ", *(finalArray + i));
}
printf("\n");

return 0;
}

/*
NAME: UniqueElementArray
FORMAL ARGUMENTS: myArray (pointer to array)
                  length (length of that array)
                  uniqueArrayLength (pointer to the integer)
RETURN TYPE: int* (address to the integer array)
POSTCONDITION: The array return the address of uniqueElementArray formed
                by dyanmic memory allocation.
SPECIFICATION: This function uniqueElementArray takes the original array and return
                array with unique element. Since, we do not know size of array until
                user input the size. We allocate the memory using dynamic memory
                allocation.

*/
int* uniqueElementArray(int* myArray, int length, int* uniqueArrayLength) {

    /*dynamically allocating memory for new array*/
    int* uniqueArray = (int*)calloc(length, sizeof(int));
    int arrayLength = 0;

    /*using for loop to access each element in the array*/
    for (int i = 0; i < length; i++) {

        /*variable to store number of occurance*/
        int counter = 0;

        /*if element is already present in uniqueArray. Do not add uniqueArray
        add if the element is not present*/
        for(int j = 0; j < length; j++){

            /*checking if element is unique*/
            if (i != j) {
                if (*(myArray + i) == *(myArray + j)) {
                    counter = counter + 1;
                }
            }
        }
    }
}

```

```

    }
    /*Performing the following operation if the operation is unique*/
    if (counter == 0) {

        *(uniqueArray + arrayLength) = *(myArray + i);
        arrayLength++;

    }
}
/*passing the length of new array with unique element to main function*/
*(uniqueArrayLength) = arrayLength;
return uniqueArray;
}

```

Method 2:

```

//This same problem can be done easier with array if suppose maaximum
// number of element in the array is constant(like 100 or 50)

/*for Visual stusio 2019 to use scanf with out error*/
#define _CRT_SECURE_NO_WARNINGS
/*
*
* AUTHOR: Bibek Dhungana
* FILENAME: lab4.c
* DATE: 2/26/2021
* SPECIFICATION: This program prints out the unique numbers in a array.
* FOR: CS 1412 Programming Principle 2 Section 504
*/

/*Importing all the required library*/
#include <stdio.h>
#include <stdlib.h>

/*defining maximum size for the array*/
#define ARRAYSIZE 50

int main1(void) {

    /*Initializing all the required variables*/

```

```

int myArray[ARRAYSIZE];
int arrayLength;

/*Printing the information to the user*/
printf("Print all unique elements of an array:\n");
printf("-----\n");

/*asking the user to enter the length of array*/
printf("Input the number of elements to be stored in the array:");
scanf("%d", &arrayLength);

/*taking input from the user and storing in the array myArray*/
printf("Input %d elements in the array:\n", arrayLength);
for (int i = 0; i < arrayLength; i++) {
    printf("element-%d:", i);
    scanf("%d", &myArray[i]);
}


/*Printing out the unique element of the array*/
printf("The unique elements found in the array are:\n");
for (int i = 0; i < arrayLength; i++)
{
    int repeatedCounter = 0;
    for (int j = 0; j < arrayLength + 1; j++)
    {
        /*search for the duplicate*/
        if (i != j) {

            /*checking if duplicate exists*/
            if (myArray[i] == myArray[j])
            {
                repeatedCounter += 1;
            }
        }
    }
    /*Printing out the unique element in the array*/
    if (repeatedCounter == 0)
    {
        printf("%d ", myArray[i]);
    }
}

return (0);
}

```

OUTPUT:

 Output.txt - Notepad

File Edit Format View Help

EXAMPLE 1:

INPUT:

Print all unique elements of an array:

Input the number of elements to be stored in the array:4

Input 4 elements in the array:

element-0:2

element-1:2

element-2:4

element-3:3

OUTPUT:

The unique elements found in the array are:

4 3

EXAMPLE 2:

INPUT:

Print all unique elements of an array:

Input the number of elements to be stored in the array:2

Input 2 elements in the array:

element-0:1

element-1:223

OUTPUT:

The unique elements found in the array are:

1 223

EXAMPLE 3:

INPUT:

Print all unique elements of an array:

Input the number of elements to be stored in the array:10

Input 10 elements in the array:

element-0:1

element-1:1

element-2:2

element-3:2

element-4:3

element-5:3

element-6:4

element-7:4

element-8:5

element-9:6

OUTPUT:

The unique elements found in the array are:

5 6

