



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN**



Diseño de algoritmos

Unidad 6. Desarrollo de programas

Actividad 3

Grupo: 8192

Clave: 2125

Semestre: 1º

Alumno:
Mauricio Luna Bautista

Profesor:
Juan Manuel Martínez Fernández

10-May-2024

Unidad 6. Actividad 3.

A partir del estudio de la unidad, elabora un programa que cumpla el siguiente objetivo: El programa deberá solicitar al usuario 3 números, los cuales deberán ser mayores de 3 y menores de 15. El programa deberá regresar los números ordenados de mayor a menor e indicar un error si el usuario ingresa números fuera de los parámetros establecidos.

Abajo se muestra el código del programa:

```
/*El programa deberá solicitar al usuario 3 números, los cuales deberán ser mayores
de 3
y menores de 15. El programa deberá regresar los números ordenados de mayor a menor
e indicar
un error si el usuario ingresa números fuera de los parámetros establecidos.
*/

#include "pch.h"
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <string.h>

int main()
{
    //Declaración de variables
    int numeros[3];
    int i, posicion, aux = 0;

    printf("Programa que ordena tres números introducidos por el usuario.\nNota:
Los números deben ser mayores que 3 y menores que 15.\n");

    //Solicitar tres números con restricción mayores que 3 y menores que 15

    for (i = 0; i < 3; i++)
    {
        do
        {
            printf("\nDigite el número [%i]: ", i+1);
            scanf_s("%i", &numeros[i]);
            if (numeros[i] <= 3 || numeros[i] >= 15)
            {
                printf("\nEl número debe ser mayor que 3 y menor que 15.
Intente de nuevo.");
            }
        } while (numeros[i] <= 3 || numeros[i] >= 15);
    }

    //Imprimimos los números originales para demostración
    printf("\nNúmeros originales: ");
```

```

for (int i = 0; i < 3; i++)
{
    printf("\n El elemento [%i] es: %i", i+1, numeros[i]);
}

//Ordenamiento por inserción con un ciclo for
for (i = 0; i < 3; i++)
{
    posicion = i; //Para que vaya iterando la posición//
    aux = numeros[i];

    while ((posicion >= 0) && (numeros[posicion - 1] > aux))
    {
        numeros[posicion] = numeros[posicion - 1];
        posicion--;
    }
    numeros[posicion] = aux; //Para refrescar en qué numero de la posición
vamos//
}

//Imprimimos los números ordenados para comparación
printf("\nNúmeros ordenados: ");
for (int i = 0; i < 3; i++)
{
    printf("\n El elemento [%i] ordenado es: %i", i + 1, numeros[i]);
}

return 0;
}

```

Conclusión

Con esta actividad fortalecí el uso de ciclos for ya que a veces puede ser confuso el iterador que inicia con 0, de igual manera el do-while para poner la restricción de los números que no se encuentren en el rango permitido.

Para el ordenamiento, decidí usar el de inserción debido a que considero que es el más sencillo en líneas de código y el más simple en operaciones, ya que solo compara el número del iterador con el anterior y directamente los cambia.

Bibliografía:

- Manzano, G., Montesano, R., Zúñiga, L.F., (2023). *Diseño de algoritmos*. Universidad Nacional Autónoma de México.
- Joyanes, L., Zahonero, I. (2014). *Programación en C, C++, Java y UML*. McGraw Hill, 2da edición. (1ra edición publicada en 2009).