Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

PROGRAMACIÓN ESTRUCTURADA

FUNCIONES Y MÉTODOS DE ORDENACIÓN Y BÚSQUEDA

ANEXO - PRÁCTICA 9

Ingeniero en Software y Tecnologías Emergentes

Brayan Ivan Perez Ventura

372781

08 de Octubre del 2023 en Ensenada, Baja California

```c
void menu()
{
    int op, n, m;
    n = 15;
    m = 4;
    int Vector_1[15];
    int matrix[m][m];
    do
    {
        system("CLS");
        op = msge_menu();
        switch (op)
        {
        case 1:
            fillVectorNoRepeat(Vector_1, n, 100, 200);
            printArr("Vector llenado satisfactoriamente");
            break;
        case 2:
            fillMatrixNoRepeat(m, m, matrix, 1, 16);
            printArr("Matriz llenado correctamente");
            break;
        case 3:
            printVector(Vector_1, n);
            break;
        case 4:
            printMatrix(m, m, matrix);
            break;
        case 5:
            bubbleSort(Vector_1, n);
            printArr("Vector ordenado correctamente");
            break;
        case 6:
            findNumber(Vector_1, n);
            break;
        }
        if (op != 0)
        {
            printArr("\nPresiona ENTER para continuar... ");
            getchar();
        }
    } while (op != 0);
    printArr("Saliendo del programa\n");
}

int msge_menu()
{
    int op;
    printArr("M E N U\n");
    printArr("1.- LLENAR VECTOR\n");
    printArr("2.- LLENAR MATRIZ\n");
    printArr("3.- IMPRIMIR VECTOR\n");
    printArr("4.- IMPRIMIR MATRIZ\n");
    printArr("5.- ORDENAR VECTOR\n");
    printArr("6.- BUSCAR VALOR EN VECTOR\n");
    printArr("0.- SALIR\n");
    op = valid("Selecciona una opcion: ", 0, 6);
    return op;
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion:
```

EN LIBRERÍA:

```
You, 4 days ago | 1 author (You)
1   //*******LIBRARIES************
2   #include <stdlib.h>
3   #include <stdio.h>
4   //*******PROTOTYPE FUNCTIONS *********
5   int valid(char msge[], int ri, int rf);
6   int existElem(int vector[], int longi, int num);
7   int counter(char array[]);
8   void mayus(char array[]);
9   void reverse(char array[]);
10  void line(char array[]);
11  void descending(char array[]);
12  void reverseDescending(char array[]);
13  void noVowels(char array[]);
14  void vowels(char array[]);
15  void minus(char array[]);
16  void capital(char array[]);
17  void noSpace(char array[]);
18  int alfaSpace(char array[]);
19  void alfaSpaceValid(char array[]);
20  int palindrome(char array[]);
21  void printArr(char array[]);
22  void ask(char array[]);
23  void fillVectorNoRepeat(int vect[], int n, int ri, int rf);
24  void printArNum(char array[], int num);
25
26  //******* FUNCTIONS DEVELOPMENT *******
27  int valid(char msge[], int ri, int rf)···
40
41  int existElem(int vector[], int longi, int num)···
54
55  int counter(char array[])···
65
66  void mayus(char array[])···
82
83  void reverse(char array[])···
102
103 void line(char array[])···
115
116 void descending(char array[])···
130
131 void reverseDescending(char array[])      You, 4 days ago • Added New Program With a
145
146 void noVowels(char array[])···
163
164 void vowels(char array[])···
178
179 void minus(char array[])···
195
196 void capital(char array[])···
234
235 void noSpace(char array[])···
246
247 int alfaSpace(char array[])···
313
314 void alfaSpaceValid(char array[])···
339
340 int palindrome(char array[])···
355
356 void printArr(char array[])···
366
367 void ask(char array[])···
372
373 void printVector(int vector[], int n)···
381
382 void fillVectorNoRepeat(int vect[], int n, int ri, int rf)···
397
398 void fillMatrixNoRepeat(int m, int n, int matrix_1[][n], int ri, int rf)···
412
413 void printMatrix(int n, int m, int matrix[][m])···
424
425 void bubbleSort(int vector[], int n)···
442
443 void printArNum(char array[], int num)···
```

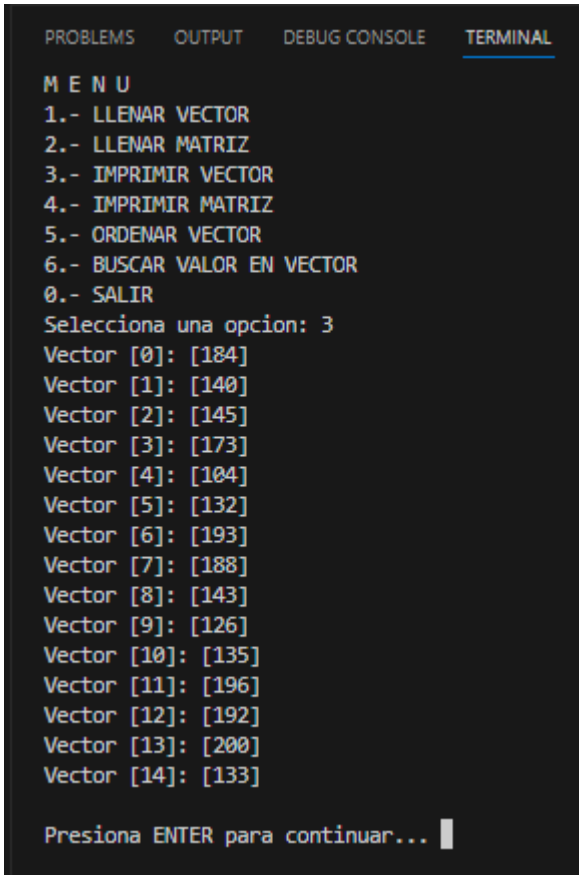
```
382  void fillVectorNoRepeat(int vect[], int n, int ri, int rf)
383  {
384      int num, range;
385      int i;
386      num = 0;
387      range = (rf - ri) + 1;
388      for (i = 0; i < n; i++)
389      {
390          do
391          {
392              num = (rand() % range) + ri;
393          } while (existElem(vect, i, num) != -1);
394          vect[i] = num;
395      }
396  }
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   P

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 1
Vector llenado satisfactoriamente
Presiona ENTER para continuar...
```

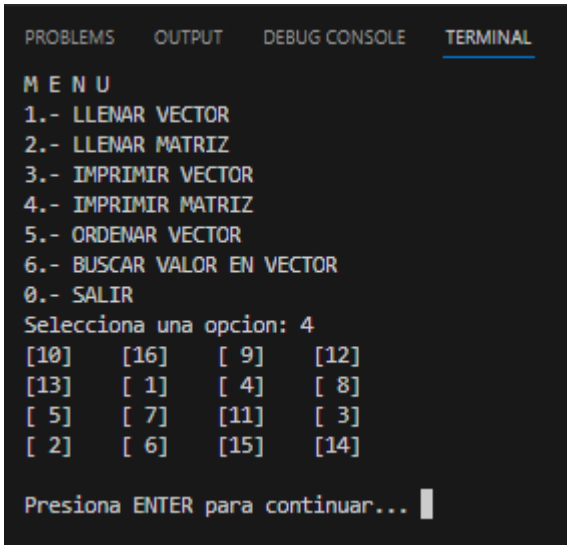
```
398  void fillMatrixNoRepeat(int m, int n, int matrix_1[][n], int ri, int rf)
399  {
400      int vector[m * n];
401      fillVectorNoRepeat(vector, m * n, ri, rf);
402
403      int k = 0;
404      for (int i = 0; i < m; i++)
405      {
406          for (int j = 0; j < n; j++)
407          {
408              matrix_1[i][j] = vector[k++];
409          }
410      }
411  }
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 2
Matriz llenado correctamente
Presiona ENTER para continuar...
```

```c
void printVector(int vector[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        printf("Vector [%d]: [%2d]\n", i, vector[i]);
    }
}
```



```c
void printMatrix(int n, int m, int matrix[][m])
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            printf("[%2d]\t", matrix[i][j]);
        }
        printf("\n");
    }
}
```



```
425  void bubbleSort(int vector[], int n)
426  {
427      int i, j;
428      int temp;
429      for (i = 0; i < n - 1; i++)
430      {
431          for (j = i + 1; j < n; j++)
432          {
433              if (vector[j] < vector[i])
434              {
435                  temp = vector[i];
436                  vector[i] = vector[j];
437                  vector[j] = temp;
438              }
439          }
440      }
441  }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMIN

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 5
Vector ordenado correctamente
Presiona ENTER para continuar...
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMIN

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 3
Vector [0]: [104]
Vector [1]: [126]
Vector [2]: [132]
Vector [3]: [133]
Vector [4]: [135]
Vector [5]: [140]
Vector [6]: [143]
Vector [7]: [145]
Vector [8]: [173]
Vector [9]: [184]
Vector [10]: [188]
Vector [11]: [192]
Vector [12]: [193]
Vector [13]: [196]
Vector [14]: [200]

Presiona ENTER para continuar...
```

```c
void findNumber(int Vector_1[], int n)
{
    int valueSearch, elementExist;
    valueSearch = valid("Ingresa el valor que desees buscar en el vector (numeros entre 100 y el 200):  ", 100, 200);
    elementExist = existElem(Vector_1, n, valueSearch); //It returns i value or -1
    if (elementExist != -1)
    {
        printArNum("El numero ha sido encontrado exitosamente, esta en la posicion: %d ", elementExist);
    }
    else
    {
        printArr("El numero introducido no esta en el vector");
    }
}       You, 4 days ago • modified:   PVBI_Act9_0_932/PVBI_Act9_1_932.c …
```

```c
int existElem(int vector[], int longi, int num)
{
    int i;

    for (i = 0; i < longi; i++)
    {
        if (vector[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 6
Ingresa el valor que desees buscar en el vector (numeros entre 100 y el 200):  199
El numero introducido no esta en el vector
Presiona ENTER para continuar...
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR
Selecciona una opcion: 6
Ingresa el valor que desees buscar en el vector (numeros entre 100 y el 200):  196
El numero ha sido encontrado exitosamente, esta en la posicion: 13
Presiona ENTER para continuar...
```