

```

// Pedro Carlos Balderrama Campos
//
// Fecha: 24 de marzo del 2024
// PCBC_ACT8_01.cpp
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
// Valores constantes
#define te 10
#define rango_minimo 30
#define rango_maximo 70
// Vectores globales
int vec1[te];
int vec2[te];
int vec3[te * 2];
int mat[4][5];
// Prototipos
void pdr_printOptions();
void clear();
void menu();
int evaluarRango(int x);
void pause();
// Prototipos especificos
void vec_manual();
void vec_random();
void vec_fusion();
void vec_print_all();
void mat_fill();
void mat_print();
void imprimirVector3();
// main
int main()
{
    int a, b;
    srand(time(NULL));
    menu();
    return 0;
}
//
void pdr_printOptions()
{
    clear();
    printf("\n1.- Llenar el vector 1 manualmente\n");
    printf("2.- Llenar el vector 2 aleatoriamente\n");
    printf("3.- Llenar el vector 3 con vector 1 y 2\n");
    printf("4.- Imprimir vectores\n");
    printf("5.- Llenar matriz 4x5\n");
    printf("6.- Imprimir matriz\n");
}

```

```

        printf("0.- Salir\n");
    }
// menu
void menu()
{
    char op[5];
    int menuOption = 0;
    do
    {
        pdr_printOptions();
        printf("\nIngresa una opcion: ");
        fflush(stdin);
        gets(op);
        menuOption = atoi(op);
        switch (menuOption)
        {
            case 1:
                vec_manual();
                break;
            case 2:
                vec_random();
                break;
            case 3:
                vec_fusion();
                imprimirVector3();
                break;
            case 4:
                vec_print_all();
                break;
            case 5:
                mat_fill();
                break;
            case 6:
                mat_print();
                break;
            case 0:
                break;
            default:
                printf("\nOpción no válida\n");
        }
    } while (menuOption != 0);
}
// Funciones
void pdr_stringCopyTo1(int a[te])
{
    for (int i = 0; i < te; i++)

```

```

    {
        vec1[i] = a[i];
    }
}
void pdr_stringCopyTo2(int a[te])
{
    for (int i = 0; i < te; i++)
    {
        vec2[i] = a[i];
    }
}
void vec_fusion()
{
    int i, j = 0;
    for (i = 0; i < te; i++)
    {
        vec3[i] = vec1[i];
    }
    for (; i < te * 2; i++)
    {
        vec3[i] = vec2[j];
        j++;
    }
}
void vec_manual()
{
    clear();
    char num[5];
    int vec_man[te];
    printf("\nNota: los numeros tienen que estar en un rango entre 30 y 70\n\n");
    for (int i = 0; i < te; i++)
    {
        printf("Escribe el valor #%d del vector: ", i + 1);
        // Lo que vimos en clase para convertir char --> entero
        fflush(stdin);
        gets(num);
        vec_man[i] = atoi(num);
        //
        if (evaluarRango(vec_man[i]) != 1)
        {
            printf("\nNo escribiste un numero que estuviera dentro del rango\n");
            i--;
        }
    }
    printf("\nSe capturo el vector con exito!\n");
    pause();
    pdr_stringCopyTo1(vec_man);
}

```

```

void vec_random()
{
    clear();
    printf("\n");
    int vec_ran[te], unique;
    for (int i = 0; i < te; i++)
    {
        do
        {
            unique = 1;
            vec_ran[i] = 1 + rand() % (21 - 1);
            for (int j = 0; j < i; j++)
            {
                if (vec_ran[j] == vec_ran[i])
                {
                    unique = 0;
                    break;
                }
            }
        } while (!unique);
        printf("El valor #%d es igual a: %d\n", i + 1, vec_ran[i]);
    }
    pause();
    pdr_stringCopyTo2(vec_ran);
}

void vec_print_all()
{
    clear();
    for (int i = 0; i < te; i++)
    {
        printf("El valor #%d del vector #1 es: %d\n", i + 1, vec1[i]);
    }
    printf("\n");
    for (int i = 0; i < te; i++)
    {
        printf("El valor #%d del vector #2 es: %d\n", i + 1, vec2[i]);
    }
    printf("\n");
    for (int i = 0; i < te * 2; i++)
    {
        printf("El valor #%d del vector #3 es: %d\n", i + 1, vec3[i]);
    }
    pause();
}

void mat_filll()
{
    clear();
    int l = 0;

```

```

    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (l < te * 2)
            {
                mat[i][j] = vec3[l];
                l++;
            }
        }
    }
    printf("Matriz de 4x5 llenada con exito!");
    pause();
}

void mat_print()
{
    clear();
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
    pause();
}

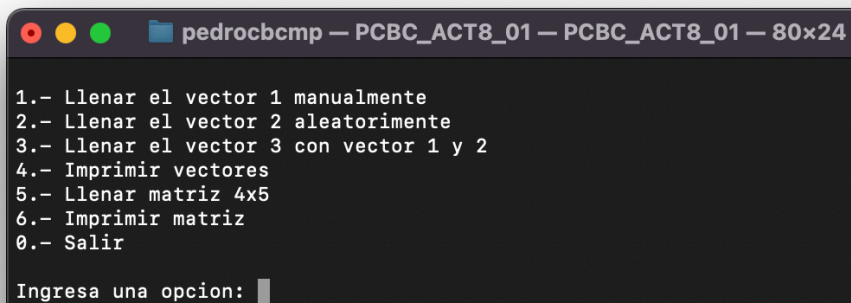
// Prototipos menos importantes
int evaluarRango(int x)
{
    if (x >= rango_minimo && x <= rango_maximo)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void clear()
{
    system("clear");
}

void pause()
{
    fflush(stdin);
    printf("\nPresiona Enter para continuar...");
    getchar();
}

```

```
}  
void imprimirVector3()  
{  
    clear();  
    for (int i = 0; i < te * 2; i++)  
    {  
        printf("El valor #%d es: %d\n", i + 1, vec3[i]);  
    }  
    pause();  
}
```



A terminal window titled "pedrobcmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x24" displays a menu of options. The options are listed as follows:

- 1.- Llenar el vector 1 manualmente
- 2.- Llenar el vector 2 aleatoriamente
- 3.- Llenar el vector 3 con vector 1 y 2
- 4.- Imprimir vectores
- 5.- Llenar matriz 4x5
- 6.- Imprimir matriz
- 0.- Salir

Below the menu, the prompt "Ingresa una opcion:" is followed by a cursor.

```
pedrocbcmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x24

Nota: los numeros tienen que estar en un rango entre 30 y 70

warning: this program uses gets(), which is unsafe.
Escribe el valor #1 del vector: 29

No escribiste un numero que estuviera dentro del rango
Escribe el valor #1 del vector: 30
Escribe el valor #2 del vector: 31
Escribe el valor #3 del vector: 32
Escribe el valor #4 del vector: 33
Escribe el valor #5 del vector: 34
Escribe el valor #6 del vector: 70
Escribe el valor #7 del vector: 71

No escribiste un numero que estuviera dentro del rango
Escribe el valor #7 del vector: 67
Escribe el valor #8 del vector: 68
Escribe el valor #9 del vector: 45
Escribe el valor #10 del vector: 46

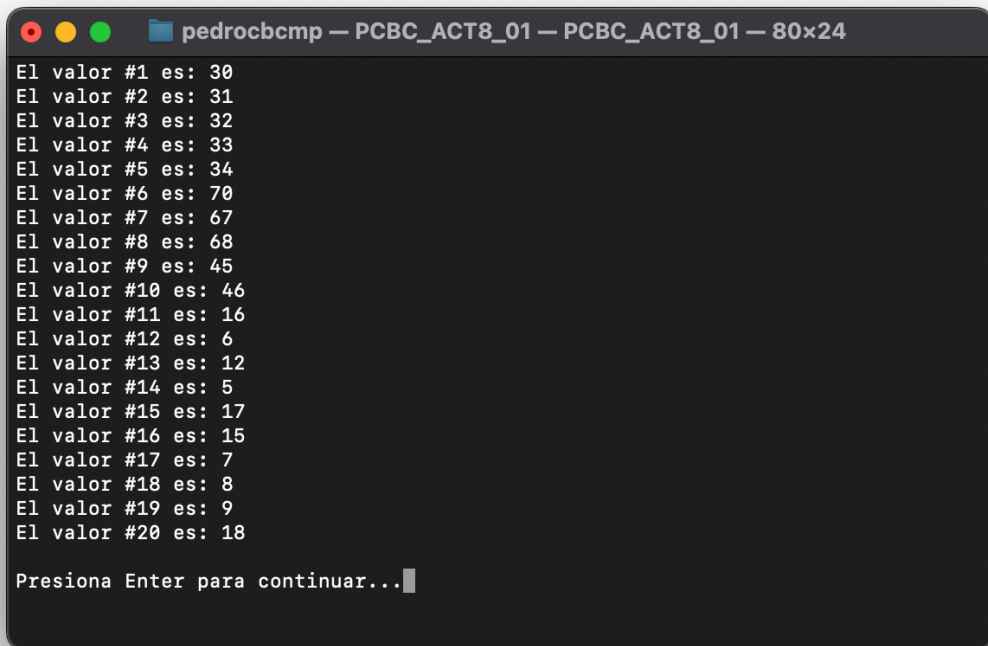
Se capturo el vector con exito!

Presiona Enter para continuar...
```

```
pedrocbcmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x24

El valor #1 es igual a: 16
El valor #2 es igual a: 6
El valor #3 es igual a: 12
El valor #4 es igual a: 5
El valor #5 es igual a: 17
El valor #6 es igual a: 15
El valor #7 es igual a: 7
El valor #8 es igual a: 8
El valor #9 es igual a: 9
El valor #10 es igual a: 18

Presiona Enter para continuar...
```



A terminal window titled "pedrobcmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x24". The window contains a list of 20 values, each labeled "El valor #X es:" followed by a number. The values are: 30, 31, 32, 33, 34, 70, 67, 68, 45, 46, 16, 6, 12, 5, 17, 15, 7, 8, 9, and 18. At the bottom, there is a prompt "Presiona Enter para continuar..." followed by a cursor.

```
El valor #1 es: 30
El valor #2 es: 31
El valor #3 es: 32
El valor #4 es: 33
El valor #5 es: 34
El valor #6 es: 70
El valor #7 es: 67
El valor #8 es: 68
El valor #9 es: 45
El valor #10 es: 46
El valor #11 es: 16
El valor #12 es: 6
El valor #13 es: 12
El valor #14 es: 5
El valor #15 es: 17
El valor #16 es: 15
El valor #17 es: 7
El valor #18 es: 8
El valor #19 es: 9
El valor #20 es: 18

Presiona Enter para continuar...
```



```
pedrocbcmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x46

El valor #1 del vector #1 es: 30
El valor #2 del vector #1 es: 31
El valor #3 del vector #1 es: 32
El valor #4 del vector #1 es: 33
El valor #5 del vector #1 es: 34
El valor #6 del vector #1 es: 70
El valor #7 del vector #1 es: 67
El valor #8 del vector #1 es: 68
El valor #9 del vector #1 es: 45
El valor #10 del vector #1 es: 46

El valor #1 del vector #2 es: 16
El valor #2 del vector #2 es: 6
El valor #3 del vector #2 es: 12
El valor #4 del vector #2 es: 5
El valor #5 del vector #2 es: 17
El valor #6 del vector #2 es: 15
El valor #7 del vector #2 es: 7
El valor #8 del vector #2 es: 8
El valor #9 del vector #2 es: 9
El valor #10 del vector #2 es: 18

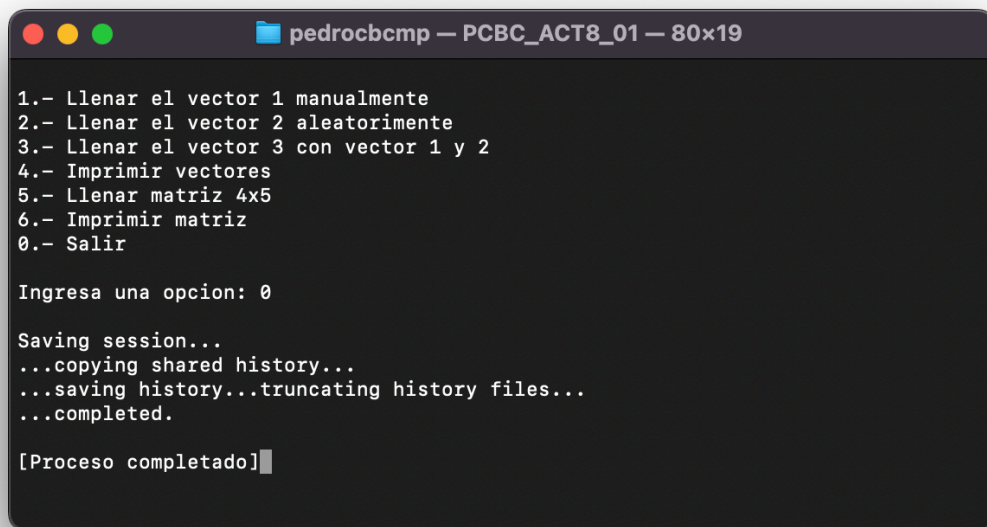
El valor #1 del vector #3 es: 30
El valor #2 del vector #3 es: 31
El valor #3 del vector #3 es: 32
El valor #4 del vector #3 es: 33
El valor #5 del vector #3 es: 34
El valor #6 del vector #3 es: 70
El valor #7 del vector #3 es: 67
El valor #8 del vector #3 es: 68
El valor #9 del vector #3 es: 45
El valor #10 del vector #3 es: 46
El valor #11 del vector #3 es: 16
El valor #12 del vector #3 es: 6
El valor #13 del vector #3 es: 12
El valor #14 del vector #3 es: 5
El valor #15 del vector #3 es: 17
El valor #16 del vector #3 es: 15
El valor #17 del vector #3 es: 7
El valor #18 del vector #3 es: 8
El valor #19 del vector #3 es: 9
El valor #20 del vector #3 es: 18

Presiona Enter para continuar...
```

```
pedrocbmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x5
Matriz de 4x5 llenada con exito!
Presiona Enter para continuar...
```

```
pedrocbmp — PCBC_ACT8_01 — PCBC_ACT8_01 — 80x19
30 31 32 33 34
70 67 68 45 46
16 6 12 5 17
15 7 8 9 18

Presiona Enter para continuar...
```



```
1.- Llenar el vector 1 manualmente
2.- Llenar el vector 2 aleatoriamente
3.- Llenar el vector 3 con vector 1 y 2
4.- Imprimir vectores
5.- Llenar matriz 4x5
6.- Imprimir matriz
0.- Salir

Ingresa una opcion: 0

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Proceso completado]
```