

Clase 1

13/03/2023

Lenguaje C: *Práctico.*

Ejercicio 1:

Realizar un programa que dado un vector de 20 posiciones se cargue mediante una función llamada cargar con datos aleatorios entre 1 y 200.

Se necesita desgranar (abrir) ese vector en otros 2 vectores que van a tener los números primos y los números compuestos.

Se tienen que llamar vecPrim y vecComp en una función llamada desgranar.

Para poder ver la información generada, desarrollar una función llamada mostrarVector, la cual pasándole de a 1 vector a la vez. Como parámetros tendrá miVector, cantElementosVector, y un título.

Resolución:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h> // para usar la función rand y que no aparezcan errores

#define n 20 // cantidad de elementos de todos los vectores
void cargar(int []);
void desgranar(int [], int [], int []);
void mostrarVector(int []);

int main()
{
    int vecPrim[n] = {0};
    int vecComp[n] = {0};
    srand(time(NULL)); // función para generar números al azar (semilla aleatoria)
    int vector[n] = {0};
    cargar(vector);
    desgranar(vector, vecPrim, vecComp);
    printf("\nnúmeros primos: ");
    mostrarVector(vecPrim);
    printf("\nnúmeros compuestos: ");
    mostrarVector(vecComp);
    return 0;
}
```

```

void cargar(int vector [])
{
    for (int i = 0; i < n; i++)
    {
        vector[i] = rand()%(200-1+1)+1;
    }
}

void desgranar(int vector [],int vecPrim [],int vecComp [])
{
    for (int i = 0; i < n; i++)
    {
        if (vector[i]==0||vector[i]==1)
        {
            vecComp[i]=vector[i];
        }
        else if (vector[i]==4)
        {
            vecComp[i]=vector[i];
        }
        for (int x = 2; x <= vector[i] / 2 + 1; x++)
        {
            if (vector[i] % x == 0)
            {
                vecComp[i]=vector[i];
                break;
            }
            else if (x == vector[i] / 2 + 1)
            {
                vecPrim[i]=vector[i];
            }
            else if (vector[i] == 2)
            {
                vecPrim[i]=vector[i];
            }
        }
    }
}

void mostrarVector(int vector[])
{
    for (int i = 0; i < n; i++)
    {
        if (vector[i] != 0)
        {
            printf("%d, ",vector[i]);
        }
    }
}

```

Ejercicio 2:

```
int v[10] = {10,20,30,40,50,60,70,80,90,100};
```

Mostrar el vector v[10] normal, como si fuera un puntero, la dirección del vector, la dirección del puntero.

Mostrar el código en hexa con %X

Resolución:

```
#include <stdio.h>
int main()
{
    int v[10] = {10,20,30,40,50,60,70,80,90,100};

    // impresion del vector "normal"

    printf("vector v: ");
    for (int i = 0; i < 10; i++)
    {
        printf("%d, ", v[i]);
    }

    // impresion del vector "como si fuera un puntero"

    int *ptr = v; // ptr apunta a la dirección de memoria del primer elemento de v
    printf("\nvector v (usando un puntero): ");
    for(int i = 0; i < 10; i++)
    {
        printf("%d ", *ptr); // se imprime el valor al que apunta ptr
        ptr++; // se incrementa el puntero para apuntar al siguiente elemento
    }

    // impresion de la dirección de memoria de cada elemento del vector.

    printf("\ndirección de memoria del vector v: ");
    for(int i = 0; i < 10; i++)
    {
        printf("%X, ", &v[i]);
    }

    // impresion de la dirección del puntero

    printf("\ndirección de memoria del puntero ptr: %X", &ptr);
    return 0;
}
```

Ejercicio 3:

Dada la siguiente estructura:

```
#include <stdio.h>
#define n 5
typedef struct
{
    int codRubro;
    char nomRubro[25];
    float precio;
} registro;
void mostrar(registro[]);
void ordenar(registro[]);
int main()
{
    registro rubro[n]={{1,"Automotor",15.25},
                       {2,"Oficinas",12.46},
                       {3,"Profesionales",14.80},
                       {4,"Escobas",48.34},
                       {5,"Servicios",10.54},};

    return 0;
}
```

Ordenar por nombre de rubro y mostrar

Resolución:

```
#include <stdio.h>
#include <string.h>
#define n 5
typedef struct
{
    int codRubro;
    char nomRubro[25];
    float precio;
} registro;
void mostrar(registro[]);
void ordenar(registro[]);
int main()
{
    registro rubro[n]= {{1,"Automotor",15.25},
                       {2,"Oficinas",12.46},
                       {3,"Profesionales",14.80},
                       {4,"Escobas",48.34},
```

```

        {5,"Servicios",10.54},};

ordenar(rubro);
mostrar(rubro);
return 0;
}

void ordenar(registro rubro[])
{
    int i, j, min_idx;
    registro temp;

    // se itera sobre todos los elementos del arreglo
    for (i = 0; i < n-1; i++)
    {
        // se encuentra el elemento mínimo en el resto del arreglo
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (strcmp(rubro[j].nomRubro, rubro[min_idx].nomRubro) < 0)
                min_idx = j;

        // se intercambia el elemento mínimo con el primer elemento
        temp = rubro[i];
        rubro[i] = rubro[min_idx];
        rubro[min_idx] = temp;
    }
}

void mostrar(registro rubro[])
{
    int i;
    printf("%-10s %-25s %s\n", "Codigo", "Nombre de rubro", "Precio");
    for (i = 0; i < n; i++)
    {
        printf("%-10d %-25s %.2f\n", rubro[i].codRubro, rubro[i].nomRubro, rubro[i].precio);
    }
}

```