Sigifredo Escobar Gómez 1128422444

Ana Lucía Castillo Ramirez 1094902206

**En un vector almacenado en la memoria, halle:**

1. Cuantos son mayores que el primer elemento

f. Cuantos son impares.

**En dos vectores almacenados en memoria, halle:**

j. Cuantos elementos hay repetidos en cada uno de los vectores

1. Descripción de alto nivel:

#include<stdio.h>

int mayores(int \* v, int n); #prototipo de la función mayores

int main()//Función principal

{

#define N 8 //Tamaño del vector

int v[] = {6, 2, 3, 4, 5, 6, 7, 8}; //El vector

printf("Los números mayores al primer elemento son: %d\n", mayores(v, N)); // Mostramos resultados

return 0; // Fin de la función

}

int mayores(int \* v, int n) //Función para encontrar los mayores al primer elemento

{

int i = 0;

int cont = 0;

while(i++ < n)

if(v[i] > v[0])

cont++;

return cont;

}

j. #include<stdio.h>

#define N 8 //Tamaño del vector

#define M 9 //Tamaño del vector

int v1[] = {1,2,2,4,2,4,7,1}; //Vector 1

int v2[] = {1,2,3,4,5,6,7,9,9};//Vector 2

int repetidos(int \* v, int n); //Prototipo de la función

int main() //Función Principal

{

printf("El el primer vector hay %d números repetidos\n", repetidos(v1, N)); //Muestra

//El resultado de los elementos repetidos en el vector uno

printf("El el segundo vector hay %d números repetidos\n", repetidos(v2, M)); //Muestra

//El resultado de los elementos repetidos en el vector dos

return 0; //Termina la función

}

int repetidos(int \* v, int n) //Función repetidos, encuentra el número de elementos

//repetidos en el vector

{

int i = 0, j = 0, k , rep = 0;

for(i = 0; i < n; i++)

for(j = i+1; j < n; j++)

if(v[i] == v[j] && i != j && i !=-1)

{

for(k = j+1; k < n; k++)

if(v[i] == v[k])

v[k] = -1;

rep++;

}

return rep; //Retorna el número de elementos repetidos en el vector.

}

f. #include<stdio.h>

#define N 9 //Tamaño del vector

int main() //Función principal

{

int impares = 0;

int i = 0;

int n[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

loop: //Ciclo que cuenta los números impares

if((n[i++] & 1) == 1)

impares++;

if(i < N)

goto loop;

printf("NÃºmeros impares: %d\n", impares); //Imprime el resultado

return 0; //Termina la función principal

}