|  | **Carrera:** Técnico Universitario en Programación  **Materia:** Laboratorio de Computación I |
| --- | --- |

**TP Funciones**

**Los siguientes ejercicios se pueden resolver utilizando las funciones del archivo funciones.h. Si se considera necesario, pueden agregarse otras funciones para resolverlos.**

1. Hacer un programa para cargar un vector de números enteros de 10 elementos.

Calcular e informar:

- El máximo valor del vector.

- La primera posición en la que aparece el máximo valor dentro del vector.

- La cantidad de veces que aparece el máximo valor dentro del vector.

1. Hacer un programa para cargar un vector de números enteros de 10 elementos.

Calcular e informar:

- El mínimo valor del vector.

- La primera posición en la que aparece el mínimo valor dentro del vector.

- La cantidad de veces que aparece el mínimo valor dentro del vector.

1. Hacer un programa para cargar un vector de números enteros de 5 elementos, de manera aleatoria con números entre el 1 y el 6

Calcular e informar:

- La cantidad de veces que aparece cada número del 1 al 6 dentro del vector.

- La suma de los valores contenidos en el vector.

- La cantidad de veces que aparece el máximo valor dentro del vector.

1. Hacer un programa para cargar un vector de números enteros de 5 elementos, de manera aleatoria con números entre el 1 y el 6

Calcular e informar:

- Copiar el vector en otro vector de igual tamaño

- Mostrar el segundo vector ordenado de menor a mayor (ascendente).

- Informar si el vector original estaba ordenado de manera ascendente.

1. Hacer un programa para cargar un vector de números enteros de 6 elementos, de manera aleatoria con números entre el 1 y el 6

Calcular e informar:

- La cantidad de valores pares contenidos en el vector

- La cantidad de valores impares.

- La cantidad de unos.

1. Hacer un programa para cargar un vector de números enteros de 15 elementos, de manera aleatoria con números entre el 1 y el 6

Calcular e informar:

- La cantidad de veces que aparece cada número del 1 al 6 dentro del vector.

Resolver el ejercicio utilizando un vector de 6 posiciones, utilizando cada posición para guardar la cantidad de veces que se repite el número +1. (Aclaración: la posición 0 del vector debe contener la cantidad de unos, la posición 1 la cantidad de dos, etc).

* Mostrar sólo las cantidades de los números que hayan salido.

1. Teniendo como referencia la función ordenarVector(), hacer otra función de nombre ordenarVectorDescendente() que ordene el vector que recibe de mayor a menor.
2. Modificar las funciones dadas para resolver los ejercicios 1 y 2 con vectores de números reales (tipo float).
3. Modificar el ejercicio 6 de manera tal que el cálculo se haga 30 veces (se carga y analiza el vector 30 veces). Informar el resultado al final.
4. Analizar qué sucede si el ejercicio anterior se repite incrementado la cantidad de veces que se carga y procesa el vector.

Funciones del archivo [funciones.h](https://github.com/mdirube/Lab1.2020.2C/blob/master/funciones.h)

| #ifndef FUNCIONES\_H\_INCLUDED  #define FUNCIONES\_H\_INCLUDED  /\* Usar en el main para que funcionen.  #include<iostream> #include<cstdlib> #include<cstdio> #include<ctime> \*/  */// DECLARACIÓN/PROTOTIPO DE FUNCIONES* void cargarVector(int v[], int tam);*///asigna valores a cada una de las posiciones del vector*  void mostrarVector(int v[], int tam);*/// muestra lo que contiene cada una de las posiciones del vector*  void ponerCero(int v[], int tam);*/// pone en 0 todas las posiciones del vector*  int posicionNumeroEnVector(int v[],int tam, int numero);*/// devuelve la posición que ocupa un número que se //envía como parámetro. Devuelve la primera posición que encuentra el número. Si no lo encuentra devuelve -1* int contarNumerosRepetidos(int v[], int numero, int tam);*///cuenta la cantidad de veces que se repite un //número en el vector*  int maximoVector(int v[], int tam);*///devuelve la posición (primera) que ocupa el máximo en el vector*  int minimoVector(int v[], int tam);*///devuelve la posición (primera) que ocupa el mínimo en el vector*  void cargarAleatorio(int v[], int tam, int limite);*///asigna valores aleatorios a cada al vector*  void ordenarVector(int v[], int tam );*/// ordena de menor a mayor un vector*  int sumarVector(int v[], int tam ); */// suma los valores contenidos en el vector*  void copiarVector(int v[], int v2[],int tam ); *///copia el vector v en v2*  bool compararVectores(int v[], int v2[],int tam );*// compara los dos vectores que recibe. Si son iguales ///devuelve true, si no devuelve false*  */// Desarrollo de funciones / Definiciones* int posicionNumeroEnVector(int v[],int tam, int numero){  int i;  for(i=0;i<tam;i++){  if(v[i]==numero) return i;  }  return -1; } int minimoVector(int v[], int tam){  int i, posMin=0;  for(i=1;i<tam;i++){  if(v[i]<v[posMin]) {  posMin=i;  }  }  return posMin; }  void ordenarVector(int v[], int tam ){  int i,j, posmin, aux;  for(i=0;i<tam-1;i++){  posmin=i;  for(j=i+1;j<tam;j++){  if(v[j]<v[posmin]) posmin=j;  }  aux=v[i];  v[i]=v[posmin];  v[posmin]=aux;  } }  int maximoVector(int v[], int tam){  int i, posMax=0;  for(i=1;i<tam;i++){  if(v[i]>v[posMax]) {  posMax=i;  }  }  return posMax; }  void cargarAleatorio(int v[], int tam, int limite){  int i;  srand(time(NULL));  for( i=0; i<tam; i++ ){  v[i]=(rand()%limite)+1;  } }  void mostrarVector(int v[], int tam){  int i;  for(i=0;i<tam;i++){  cout<<v[i]<<"\t";  } }  int contarNumerosRepetidos(int v[], int numero, int tam){  int i, cant=0;  for(i=0;i<tam;i++){  if(v[i]==numero) cant++;  }  return cant; }  void ponerCero(int v[], int tam){  int i;  for(i=0;i<tam;i++){  v[i]=0;  }  }  int sumarVector(int v[], int tam ){  int i, suma=0;  for(i=0;i<tam;i++){  suma+=v[i];  }  return suma; }  void copiarVector(int v[], int v2[],int tam ){  int i;  for(i=0;i<tam;i++){  v2[i]=v[i];  } }  bool compararVectores(int v[], int v2[],int tam ){  int i;  for(i=0;i<tam;i++){  if(v2[i]!=v[i]) return false;  }  return true; }  void cargarVector(int v[], int tam){  int i;  for(i=0;i<tam;i++){  cout<<"INGRESE NUMERO: ";  cin>>v[i];  }  } |
| --- |