**Clase 1**

**Ejercicio 1-1:**

Ingresar números enteros, hasta que el usuario lo decida. Calcular y mostrar:

1. El promedio de los números positivos.
2. El promedio de los números negativos.

**Ejercicio 1-2:**

Ingresar 5 números. Determinar cuál es el máximo y el mínimo e indicar en qué orden fue ingresado.

**Ejercicio 1-3:**

Realizar el algoritmo que permita ingresar los datos de los alumnos de una división de la UTN FRA, los datos solicitados son:  
Legajo  
Tipo cursada(**L**: "libre" - **P**: "presencial" – **R**: "remota")  
Cantidad de materias( mas de cero y menos de 8)  
Sexo (**F**: “femenino” – **M**: “masculino” , **O**: “no binario”)  
Nota promedio (entre 0 y 10)  
Edad (validar)  
***Se debe informar de existir, o informar que no existe , en el caso que corresponda.***a) El Legajo del mejor promedio femenino.  
b) El legajo del más joven de los alumnos masculinos entre los que dan libre.  
c) El promedio de nota por sexo.  
d) La edad y legajo del que cursa más materias. (Sin importar su género)

**Clase 2**

**Ejercicio 2-1:**

Crear una función que reciba como parámetro un numero entero. La función retornara 1 en caso de que sea positivo, -1 en caso de que sea negativo y 0 en caso de que sea 0. Realizar la prueba lógica de la función en el main.

**Ejercicio 2-2:**

Realizar un programa que permita el ingreso de 10 números enteros. Determinar el promedio de los positivos, la cantidad de ceros y del menor de los negativos la suma de los antecesores.

Nota:

Utilizar la función del punto anterior y desarrollar funciones para resolver los procesos que están resaltados.

**Clase 3**

**Ejercicio 3-1 (VECTORES Y FUNCIONES):**

Pedir el ingreso de 10 números enteros entre -1000 y 1000. Determinar:

1. Cantidad de positivos y negativos.
2. Sumatoria de los pares.
3. El mayor de los impares.
4. Listado de los números ingresados.
5. Listado de los números pares.
6. Listado de los números de las posiciones impares.

Se deberán utilizar funciones y vectores.

**Ejercicio 3-2 (VECTORES Y FUNCIONES):**

Realizar un programa que permita el ingreso de 10 números enteros distintos de cero. La carga deberá ser aleatoria (todos los elementos se inicializarán en cero por default. Determinar el promedio de los positivos, y del menor de los negativos la suma de los antecesores.

Utilizar funciones y vectores.

**Ejercicio 3-3 (ORDENAMIENTO):**

Realizar un programa que permita el ingreso de 10 números enteros (positivos y negativos). Necesito generar un listado de los números positivos de manera creciente y negativos de manera decreciente. (Como máximo 4 for)

Ejemplo:

Listado 1 : 4, 5, 6, 10, 18, 29

Listado 2 : -1,-5,-9,-12

**Ejercicio 3-4 (CADENAS):**

Pedirle al usuario su nombre y apellido (en variables separadas, una para el nombre y otra para el apellido). Ninguna de las dos variables se puede modificar.

Debemos lograr guardar en una tercer variable el apellido y el nombre con el siguiente formato:

Por ejemplo:

Si el nombre es juan ignacio y el apellido es gOmEz, la salida debería ser:

**G**omez, **J**uan **I**gnacio

**Ejercicio 3-5 (VECTORES PARALELOS O ESTRUCTURAS):**

Crear la estructura Jugador (nombre, goles metidos, partidos jugados, promedio de goles).

Crear una función que permita cargar los datos de un jugador y otra que los muestre.

Una tercera función calculará el promedio de goles.

Cargar los datos de 5 jugadores. Y mostrarlos por pantalla.

**Clase 4**

**Ejercicio 4-1 (Estructuras facil):**

Crear la estructura empleado con idEmpleado, nombre, sueldo y puesto (programador, analista, tester).

Crear un menu con las siguientes opciones:

1. Cargar empleados. Carga secuencial.
2. Mostrar empleados. Ordenados por puesto.
3. Mostrar solo los empleados analistas.
4. Mostrar cuantos empleados tester hay.

**Ejercicio 4-2 (Estructuras dificil):**

Una empresa importadora que comercializa productos Apple, decide registrar de sus productos los siguientes datos:

* **idProducto** (numerico)
* **descripcion** (alfanumérico)
* **nacionalidad** (numérico, por el momento utilizaremos un define: EEUU - CHINA - OTRO)
* **tipo** (numérico, por el momento utilizaremos un define: IPHONE -MAC - IPAD - ACCESORIOS)
* **precio** (numérico decimal)

Realizar un programa que permita interactuar con un menú de usuarios con las siguientes opciones:

1. **ALTA** Producto: Se ingresan los datos de UN solo producto. Siempre y cuando haya espacio disponible en el array.
2. **BAJA** Producto: A partir del ingreso del ID. Si existe el producto desaparece de la lista, dejando espacio disponible para un nuevo producto.
3. **MODIFICACIÓN** Producto: A partir del ingreso del ID. Si existe se podrá modificar el precio o el tipo.
4. **LISTADO** Productos.
5. **LISTADO** ordenado por precio.
6. **LISTADO** ordenado por descripción.

**Clase 8 (Punteros)**

**Ejercicio 8.1**

Realizar una función Suma que reciba como parámetros dos enteros (los que va a sumar) y un puntero a entero que guardara el resultado de esta. La función retorna 1 si pudo realizar la suma y cero en caso de no haberla podido realizar.

**Ejercicio 8.2**

Realizar una función que reciba un puntero a char y dos char. La función deberá reemplazar en la cadena cada ocurrencia del primer carácter recibido, por el segundo. Retornando la cantidad de reemplazos o -1 en caso de error.

**Ejercicio 8.3**

Realizar una función que reciba como parámetros un array de enteros, el tamaño del array y un entero por referencia. La función calculara el máximo valor de ese array y utilizara el valor por referencia para retornar ese valor.

**Ejercicio 8.4**

Escriba una función que reciba dos números enteros y los devuelva ordenados. Es decir que en el primer parámetro debe devolver el menor valor y en el 2do. el mayor.

**Ejercicio 8.5**

Realizar una función que reciba como parámetros dos vectores. La función deberá determinar si el segundo vector esta contenido en el primero.

**Ejercicio 8.6 (Punteros a funciones)**

Realizar una función que respete el siguiente prototipo:

int Calculador(int, **int(\*pFunc)(int, int)**, int, int\*);

Dicha función deberá recibir dos operandos y un operador representado mediante la función pasada como parámetro. El puntero a entero, servirá como parámetro de salida y guardará el resultado de la operación.

**Ejercicio 8.7 (Punteros a funciones)**

Realizar una función que respete el siguiente prototipo:

int Ordenador(eProducto\*,int , **int(\*pFunc)(eProducto\*, eProducto\*)**);

La función Ordenador recibirá como parámetros:

1. el array de productos, dado por el puntero a eProducto
2. la cantidad de elementos del array
3. el criterio de ordenamiento, dado por la función pFunc.

Veamos un  ejemplo de la función criterio:

int CompararPorMarca(eProducto\*,eProducto\*);

int ComparaPorPrecio(eProducto\*,eProducto\*);

**Clase 9 (Memoria dinamica)**

**Ejercicio 9.1**

Realizar una función que reciba como parámetro un array de enteros, su tamaño y un entero. La función se encargará de buscar el valor  y borrará la primera ocurrencia del mismo. El array deberá reestructurarse dinámicamente.

valor = 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 4 | 2 | 1 | 3 |

0 1 2 3 4

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 4 | 1 | 3 |

0 1 2 3

**Ejercicio 9.2**

Realizar una función que reciba como parámetro un array de enteros, su tamaño, un valor entero y un índice. La función se encargará de insertar el valor entero en el índice especificado. El array deberá reestructurarse dinámicamente.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 5 | 4 | 9 |

0 1 2 3

indice=2

valor =  10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 5 | 10 | 4 | 9 |

0 1 2 3 4

**Ejercicio 9.3**

Realizar una función que retorne un puntero a una estructura. Dicha función deberá crear una variable del tipo estructura, hardcodeara datos en esa estructura y retornara un puntero a esos datos. Mostrar varias veces el dato creado en la funcion, desde el main.