

FACULTAD DE
INGENIERÍA

INGENIERÍA EN COMPUTACION
INGENIERIA DE SISTEMAS
INGENIERIA ELECTRONICA



GUÍA DE TRABAJOS PRÁCTICOS

PROGRAMACION II
Prof.: Ing. María Alejandra Bosio

2018



TRABAJO PRACTICO DE AULA F020

TRABAJOS PRÁCTICOS DE AULA

ÍNDICE

ENCUADRAMIENTO DE LA FORMACIÓN PRÁCTICA	3
CONSIGNAS GENERALES	3
INFORME FINAL DEL TRABAJO PRÁCTICO	3
TRABAJO PRACTICO N° 1 - Programación en C++. Funciones. Arreglos	4
TRABAJO PRÁCTICO N° 2 - Punteros	13
TRABAJO PRÁCTICO N° 3 - Clases y herencia	15
TRABAJO PRÁCTICO N° 4 - Archivos y Manejo de Excepciones	18
TRABAJO PRÁCTICO EVALUABLE	19

1. ENCUADRAMIENTO DE LA FORMACIÓN PRÁCTICA

La formación práctica que se desarrolla en esta GTP incluye la resolución de problemas tipo o rutinarios y de problemas abiertos de Ingeniería.

2. CONSIGNAS GENERALES

- Los problemas propuestos en la presente GTP pueden ser resueltos en forma individual o grupal.
- La resolución de los problemas se realizará en lenguaje C++.

3. INFORME FINAL DEL TRABAJO PRÁCTICO

3.1 Informe de cada Trabajo Práctico

Para cada trabajo practico los alumnos deben presentar por escrito el desarrollo de cada de los problemas indicados como obligatorios en un informe.

Este informe debe contener los siguientes ítems:

1. Objetivo de la práctica
2. Desarrollo de la práctica
3. Conclusiones

3.2 Presentación de la carpeta de trabajos prácticos

Al finalizar el cursado de la asignatura el alumno debe presentar la carpeta de los trabajos prácticos desarrollados. La misma debe contener los siguientes ítems:

1. Carátula, ver: Carátula Presentación de TP.
2. Índice con el detalle de los trabajos prácticos.
3. Los trabajos prácticos.

TRABAJO PRACTICO Nº 1 - Programación en C++

1. Objetivos de Aprendizaje

- a) .Repasar los conceptos de programación ya adquiridos en asignaturas anteriores.
- b) Adquirir habilidad en la resolución de problemas y optimización del código.

2. Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la Unidad 1 de la programación de la asignatura.

3. Consignas a desarrollar en el trabajo práctico:

I. Estructuras secuenciales, de bifurcación y repetitivas.

1. Dada una medida de tiempo expresada en horas, minutos y segundos con valores arbitrarios, obtenga un programa que transforme dicha medida en una expresión correcta. Por ejemplo, dada la medida 3h 118m 195s, el programa deberá obtener como resultado 5h 1m 15s.
2. Dado un numero verificar:
 - Que tenga dos dígitos
 - Que sus dígitos sean paresEn caso afirmativo calcular el Promedio de sus dígitos.
3. Validar el día, mes y el año para una fecha suministrada sabiendo que los años van desde 1960 hasta 2021.
4. Escriba un programa, que dada una hora del día en Horas, Minutos y Segundos calcule la hora correspondiente 15 segundos después. (Validar la entrada de datos).
5. La fecha de Pascua corresponde al primer domingo después de la primera luna llena que sigue al equinoccio de primavera, y se calcula con las siguientes expresiones:
A = año mod 19
B = año mod 4
C = año mod 7
D = (19*A + 24) mod 30
E = (2*B + 4*C + 6*D + 5) mod 7
N = (22 + D + E)
En el que N indica el número del día de marzo (o Abril si N > 31) correspondiente al domingo de Pascua. Realizar un programa que dado un año calcule esa fecha.
6. Calcular el valor de π aplicando la fórmula: $\pi = 4 * (1 - 1 / 3 + 1 / 5 - 1 / 7 + \dots)$ incluyendo hasta el término 1 / 99.

TRABAJO PRACTICO DE AULA F020

7. Obtener un programa que calcule el factorial de un número n .
8. Obtener un programa que calcule y muestre e^x utilizando la serie:
$$e^x = 1 + x/1! + x^2/2! + x^3/3! \dots + x^n/n!$$
9. Obtener un programa que lea un número natural y diga si es o no es triangular.
A saber: un número N es triangular si, y solamente si, es la suma de los primeros M números naturales, para algún valor de M .
Ejemplo: 6 es triangular pues $6 = 1 + 2 + 3$.
10. Obtener un programa que muestre la serie de Fibonacci 0, 1, 1, 2, 3, 5, 8, 13, 21, ...
Los primeros términos son 0 y 1, los siguientes son la suma de los dos anteriores.
11. Desarrollar un programa en el cual introduzca un número entero positivo e invierta los dígitos del número. Mostrar en pantalla el número invertido.
12. Solicitar al usuario un valor entero y decir si es capicúa.
13. Leer un valor entero, positivo y comprobar si contiene la cifra 3.
14. Escribir un programa que lea un número entero y una posición dentro de este número, y muestre la cifra que corresponde a la posición (suponiendo que en la posición 1 se encuentran las unidades).
15. Escribir un programa que sume las cifras de un número entero positivo.
16. El número mágico de una persona puede considerarse como la suma de las cifras de su día de nacimiento, repitiendo el proceso hasta que la suma de las cifras devuelva un número menor de 10. Leer la fecha de nacimiento de una persona, utilizando el formato aaaammdd (utilizando un único entero) y calcular su número mágico.

II. Funciones:

1. Obtener una función que calcule el M.C.D. de dos números. Generar dos soluciones, una por repetición y la otra por recursión.
2. Obtener una función que dado un numero entero calcule la cantidad de dígitos que posee.
3. Obtener una función que dados dos enteros devuelva 1 si son números amigos y 0 en caso contrario. Dos números amigos son dos enteros positivos a y b tales que a es la suma de los divisores propios de b , y b es la suma de los divisores propios de a . Por ejemplo (220, 284) son números amigos, ya que:
Los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, que suman 284.
Los divisores propios de 284 son 1, 2, 4, 71 y 142, que suman 220.

TRABAJO PRACTICO DE AULA F020

4. Obtener una función que calcule el factorial de un numero n. Generar dos soluciones, una por repetición y la otra por recursión.
5. Escribir una función que intercambie el valor de dos variables, es decir si $X=5$ e $Y=7$ tras aplicar la función, por ejemplo haciendo "intercambiar(X,Y)" se tiene que $X=7$ e $Y=5$.
6. Escribir una función "EsMultiplo" que sirva para determinar si un número es múltiplo de otro. Utilizarla en un programa que pida la antigüedad de un trabajador y calcule cuantos trienios de antigüedad tiene.
7. Escribir una función que ingrese un numero cualquiera y que devuelva como resultado la suma de sus dígitos , ejemplo si el número fuera 23 la suma seria 5.
8. Obtener una función que dado un número entero lo descomponga dígito a dígito, lo vuelva a componer al revés y lo devuelva. Por ejemplo: si le damos el número 187365, el programa mostrará 563781.

III. Arreglos:

1. Definir un vector de 20 elementos. Leer 20 números enteros positivos, indicar cual es el mayor y en que orden fue leído (a que elemento pertenece). Mostrar el vector completo.
2. Representar un polinomio completo de grado n mediante un arreglo. Ingresar el grado del polinomio y sus coeficientes. A continuación ingresar un valor x y calcular el polinomio evaluado en ese valor. Utilizar funciones en la resolución.
3. Leer los n elementos de un vector y ordenarlo de mayor a menor.
4. Una empresa de colectivos tiene 3 líneas de 12 coches cada una. Por cada viaje el chofer entrega al llegar a la terminal una planilla con el número de coche (de 1 a 12), número de línea (de 1 a 3) y la recaudación del viaje. Las planillas se entregan sin ningún orden. Se pide informar:
 - La recaudación total por línea de colectivo
 - La recaudación total por coche
 - La recaudación total general.
5. Dado un vector de N valores enteros, se desea eliminar los repetidos.
6. Leer la cantidad de elementos de un vector y los datos en él. A continuación leer un numero k y reorganizar el arreglo en modo que los datos menores a K queden delante y los iguales o mayores detrás. No se debe utilizar un algoritmo de ordenación.
7. Dado un arreglo y un numero k determinar si el numero esta presente dentro del arreglo, en ese caso dar su ubicación.

TRABAJO PRACTICO DE AULA F020

8. Implementar un programa que solicite la introducción de un vector de N elementos, invierta su contenido (intercambie el primer elemento con el ultimo, el segundo con el penúltimo, etc.) y lo muestre.
9. Implementar un programa que solicite la introducción de una matriz de NxM y a continuación muestre el máximo valor almacenado en la matriz.
10. Implementar una función que, dada una matriz m, rellene con 1s el perímetro de la matriz (elementos de la primera y última fila, y de la primera y última columna) y con 0s el resto de la matriz (elementos centrales).
11. Implementar una función que, dada una matriz m y una posición (i,j) de dicha matriz, calcule la suma de elementos adyacentes al elemento m(i,j)
12. Una matriz mágica es una matriz cuadrada (tiene igual número de filas que de columnas) que tiene como propiedad especial que la suma de las filas, las columnas y las diagonales es igual. Construir un algoritmo que compruebe si una matriz de datos enteros es mágica o no.
13. En un terreno ondulado se quiere instalar antenas y es necesario que estas ocupen un lugar prominente. Para facilitar la elección se ha cuadrículado el terreno a los efectos de hacer un modelo digital del mismo. A cada una de las cuadrículas se le asigna un valor numérico entero representativo de la altura media del terreno dentro de esa cuadrícula. Definimos que una celda es una loma de prominencia k si dentro de un cuadrado de tamaño $2k+1$ del cual la celda loma es centro, esta celda posee la mayor altura respecto de las celdas vecinas. Suponiendo que se ingresan las altura medias en una matriz A(MxN) y el valor k donde $0 < k < 10$, determinar la posición de todas las lomas .
Ejemplo:

$M = 5; N = 10; k = 1$

t :

3	4	6	2	4	6	4	2	4	6
6	9	2	8	5	9	5	2	5	8
1	8	2	2	3	8	4	6	2	3
4	9	4	5	6	7	3	2	8	1
3	6	3	8	4	1	3	7	9	5

La solución deberá indicar 2 lomas, ubicadas en:
(2, 6) y (4, 2).

14. Observe la siguiente secuencia:
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Cada número de la misma se ha obtenido como la mitad del anterior si éste era par o el triple más uno si era impar. Por ejemplo, el 34 es par, luego le sigue el 17 que es su mitad; el 5 es impar, luego le sigue su triple 15 aumentado en 1 o sea 16.

TRABAJO PRACTICO DE AULA F020

El número que encabeza la secuencia lo denominamos semilla de la misma, en este caso el 22. Independientemente de cual sea la semilla la secuencia siempre llega a 1 y esto se considera el fin de la misma. Dada una semilla se desea conocer

- Cuan larga es la secuencia
- Cuál es el número más grande que contiene

15. Dos amigas necesitan enviarse por correo electrónico mensajes muy importantes y extremadamente reservados. Ante el temor de que alguien pueda leer los correos, deciden codificar los mensajes. La estrategia que utilizan es la siguiente: ciertos trozos del texto los escriben en orden inverso y los encierran entre paréntesis, de manera tal de no olvidar que esos trozos deben ser leídos al revés. Obtener un programa al cual se ingrese el texto codificado y lo devuelva decodificado.

Ejemplo;

```
Hoy (.sh 22 sal a) (ed asac ne sominuer son) Marcelo.
```

```
Hoy a las 22 hs. nos reunimos en casa de Marcelo.
```

IV. Estructuras:

1. Escribir un programa que defina una estructura (Equipo) que permita contener los datos relativos a equipos de football.

Los datos a considerar son:

- Nombre del equipo (tira de caracteres de longitud 20)
- Cantidad de goles realizados (entero)
- Cantidad de goles recibidos (entero)

La información debe ingresarse y memorizarse en un arreglo.

Mostrar como resultado los datos completos de los equipos cuya cantidad de goles realizados supere a los recibidos.

2. Un comercio tiene a la venta una serie de productos. Para cada producto se dispone de:

- Descripción (tira de caracteres de longitud 20)
- Stock del producto (entero)
- Año de vencimiento (entero)

El programa debe leer la información anterior mediante una estructura y generar un arreglo con la misma.

Luego se debe modificar el arreglo anterior eliminando los productos vencidos.

3. En un concurso los participantes son sometidos a 10 pruebas diferentes, Cada prueba otorga un puntaje entre 0 y 10. La información del concurso debe ser representada mediante una estructura que contiene los siguientes campos:

- Nombre del participante (tira de caracteres de longitud 20)
- Una lista de 10 números enteros.

El programa debe solicitar el ingreso de los datos anteriores y determinar:

- El ganador de cada prueba
- El ganador del concurso

TRABAJO PRACTICO DE AULA F020

4. Definir un arreglo de longitud 10 donde cada elemento es una estructura "empleado" que contiene la siguiente información:
- nombre
 - dirección (calle, número, localidad)
 - edad
 - sexo (M = masculino, F = femenino)

Realizar las siguientes operaciones:

- Ingresar la información de los empleados.
- listar los empleados menores de 25 años.
- listar los empleados que viven en Avellaneda.
- listar los empleados que viven en Avellaneda menores de 25 años.
- ordenar el vector por nombre y listarlo.

5. Se dispone de la información correspondiente a un grupo de N atletas (máximo 100). Para cada uno se tiene:

- Deporte
- Número de medallas obtenidas
- Datos personales
 - Nombre
 - Fecha de nacimiento.
 - País de origen

Obtener un programa que luego se solicite el ingreso de la información antes detallada, muestre los datos del atleta que ha ganado mayor número de medallas.

6. Realizar una agenda para guardar los datos de hasta 200 personas de las cuales se toman los siguientes datos:

- Nombre
- Apellido
- Dirección
- Localidad
- Código Postal
- Fecha de nacimiento (Día, Mes y Año)

Se pide que una vez cargados los datos, se disponga de un menú de opciones que permita hacer las siguientes consultas:

- Consulta por apellido
- Consulta por localidad
- Consulta por Año de nacimiento

Realizar un programa con un menú de opciones para hacer, además de las búsquedas, altas, bajas y modificaciones (ABM) de la agenda.

7. En una biblioteca se registran los libros existentes, informándose:

- CODIGO DE LIBRO
- CANTIDAD DE EJEMPLARES
- AUTOR
- TÍTULO

TRABAJO PRACTICO DE AULA F020

Luego de registradas las existencias se visualiza un menú que permite al bibliotecario elegir alguna de las siguientes opciones:

OPCION 1: PRESTAMOS DE LIBROS

El bibliotecario debe ingresar el CODIGO DE LIBRO. El programa debe informar si la operacion de prestamo fue exitosa o no (la operacion es exitosa solo si el codigo existe y la cantidad de ejemplares es mayor a 1).

OPCION 2: DEVOLUCION DE LIBROS

El bibliotecario debe ingresar el CODIGO DE LIBRO. El programa debe informar si la operacion de devolucion fue exitosa o no (la operacion es exitosa solo si el codigo existe).

OPCION 3: FIN

Significa que no se sigue trabajando con el menú de opciones y que antes de finalizar debe mostrar un listado de los libros prestados ese día en forma decreciente por cantidad de veces que se prestó.

8. Una empresa registra los articulos en existencia informando:

- CODIGO DE ARTICULO
- DESCRIPCION DEL ARTICULO
- CANTIDAD EN STOCK
- PRECIO UNITARIO

Luego del registro del stock se debe visualizar un menú de opciones que permita al usuario elegir alguna de las siguientes opciones:

OPCION A: registrar compras y ventas , para lo cual informar :

CODIGO DE ARTICULO

CODIGO DE MOVIMIENTO (1 o 2 caso contrario se informa error y vuelve a ingresar). El código 1 indica una venta y 2 una compra.

CANTIDAD (comprada o vendida de acuerdo al código del movimiento)

OPCION B: consultar el precio unitario de un artículo.

El usuario ingresa el Código de artículo y el sistema le debe mostrar la descripción del artículo y el precio unitario.

OPCION C: consultar existencia en stock de un artículo.

El usuario ingresa el Código de artículo y el sistema le debe mostrar la descripción del artículo y la existencia.

En todas las opciones se debe verificar que el codigo de articulo ingresado exista.

OPCION D: finalizar la ejecución del programa.

TRABAJO PRÁCTICO Nº 2 - Punteros

1. Objetivos de Aprendizaje

- a) Comprender el significado de los punteros y su relacion con arreglos y cadenas.
- b) Desarrollar programas utilizando gestion dinamica de memoria..

2. Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad 3 de la programación de la asignatura. Los ejercicios 3, 5, 6, 7, 9, y 11 son de presentacion obligatoria. Los demas son de presentacion optativa.

3. Consignas a desarrollar en el trabajo práctico:

1. Realizar un programa que rellene de forma aleatoria con los primeros 100 numeros un vector de 15 compnentes. Mostrar por medio de punteros los valores en el vector y la direccion de memoria de cada uno.
2. Ingresar una cadena de caracteres de longitud maxima 10, enviarla como parametro a una función que maneje punteros de modo que retorne la misma cadena de caracteres pero en mayúsculas.
3. Realizar un programa que ingrese una cadena de caracteres de máximo 50 elementos y la envíe como parámetro a una función que maneje punteros de modo que la función invierta la cadena.
4. Ingresar dos cadenas de caracteres enviarlas como parametros a una función que utilice punteros: La función debe comparar las dos cadenas y devolver 1 si son iguales y 0 si son diferentes.
5. Obtener una función copia que utilice punteros para copiar una cadena en otra.
6. Realice un programa que permita ingresar una cadena de caracteres de máximo 50 elementos, la envíe como parametro a una función que utilizando punteros y retorne el número de vocales minúsculas que contiene la cadena.
7. Definir un vector de N compnentes enteras de modo que la gestion de memoria sea dinamica utilizando new. Ingresar los datos en el vector. Mostrar el contenido del vector en dos modos: usando notacion vectorial y luego usando aritmetica de punteros.

TRABAJO PRACTICO DE AULA F020

8. Se pide crear un programa que haciendo uso de la reserva dinámica de memoria almacene un número determinado de valores (n) obtenidos de forma aleatoria, entre 0 y 100 y los ordene de mayor a menor
9. Se pide crear un programa que pida una serie de números al usuario y halle el máximo, el mínimo y la media aritmética de ellos. Para ello se debe crear una variable puntero tipo float, pedir al usuario que introduzca el número de datos, y sucesivamente los datos a cargar en el arreglo. Recordar que se debe reservar memoria de forma dinámica.
10. Obtener un programa que permita reservar memoria dinámica para una matriz de datos reales realizando los siguientes pasos:
 - a. Crear un puntero a punteros del tipo de datos correspondiente.
 - b. Reservar memoria para el array de filas.
 - c. Hacer un bucle para reservar memoria para las columnas de cada fila:
 - d. Ingresar los datos en la matriz y mostrar su contenido usando índices y los corchetes [].
 - e. Liberar la memoria asignada.

TRABAJO PRÁCTICO Nº 3 - Clases y herencia

2. Objetivos de Aprendizaje

- a) Comprender el concepto de Orientacion a Objetos y tipos abstractos.
- b) Comprender el concepto de herencia.
- c) Construir programas en C++ utilizando definiciones de clases y herencia.

2. Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidades 4,5,6 y 7 de la programación de la asignatura.

Los ejercicios 3, 4, 7, 8, 9 y 10 son de presentacion obligatoria. Los demas son de presentacion optativa.

3. Consignas a desarrollar en el trabajo práctico:

1. Realizar una clase que permita representar una fecha.
 - Definir los datos miembros de la clase.
 - Definir si fuera necesario funciones de carga y muestra de los datos miembro.
 - Definir un constructor que inicializa la fecha a una fecha dada.
 - Definir un constructor que inicializa la fecha en 01/01/1900.
 - Definir sobrecargas de los siguientes operadores:
 - ++ y – incrementa o decrementa la fecha en 1 día.
 - + y – suma a una fecha un cierto numero de días.
 - Realizar un programa principal que haga uso de la clase.
2. Realizar una clase que permita representar un numero racional.
 - Definir los datos miembros de la clase.
 - Definir si fuera necesario funciones de carga y muestra de los datos miembro.
 - Definir uno o mas constructores para inicializar los datos miembro.
 - Definir sobrecargas de los siguientes operadores:
 - + suma 2 racionales generando otro como resultado.
 - – resta 2 racionales generando otro como resultado.
 - ++ incrementa en 1 un racional
 - — decrementa en 1 un racional.
 - +=suma un entero a un racional.
 - -=suma un entero a un racional
 - Definir ademas una función miembro que implemente la simplificacion del racional.
 - Realizar un programa principal que haga uso de la clase.
3. Realizar una clase que permita representar un número complejo.

TRABAJO PRACTICO DE AULA F020

- Definir los datos miembros de la clase.
 - Definir si fuera necesario funciones de carga y muestra de los datos miembro.
 - Definir uno o más constructores para inicializar los datos miembro.
 - Definir la sobrecarga de los operadores +, - y * para efectuar estas operaciones entre complejos.
 - Definir una función miembro para calcular el conjugado de un complejo.
 - Realizar un programa principal que haga uso de la clase.
4. Realizar una clase que permita representar un cronometro. Dicho cronometro consta de horas, minutos y segundos. Las horas no tienen límite en valor mientras que los minutos y segundos llegan al maximo hasta 59.
- Definir los datos miembros de la clase.
 - Definir si fuera necesario funciones de carga y muestra de los datos miembro.
 - Definir un constructor que inicializa a cero el coronometro.
 - Definir un metodo Reset que permita llevar a cero el cronometro.
 - Definir la sobrecarga del operador ++ para producir el incremento del cronometro en un seg.
 - Realizar un programa princpal que haga uso de la clase.
5. Realizar una clase que permita representar un punto en coordenadas cartesianas.
- Definir los datos miembros de la clase.
 - Definir si fuera necesario funciones de carga y muestra de los datos miembro.
 - Definir un constructor que inicializa el punto en el origen de coordenadas..
 - Definir la sobrecarga del operador ++ que incrementa en 1 ambas coordenadas.
 - Idem para el operador --.
 - Definir 2 sobrecargas para el operados +, una para sumar 2 puntos entre si y otra para sumar a un punto un valor entero.
 - Idem para el operador -.
 - Definir la/las función/es miembro para obtener la conversion en coordenadas polares.
 - Realizar un programa princpal que haga uso de la clase.
6. Obtener una clase que represente una cuenta bancaria. Para la misma se tiene como informacion:
- Numero de cuenta.
 - Saldo.
 - Tasa de interes
- Definir los datos miembro de la clase.
- Obtener los metodos para:
- Crear una cuenta nueva (constructor) a partir de la asignacion del número de cuenta y una tasa de interes. Toda nueva cuenta se crea con saldo 0.
 - Carga y visualizacion de los datos miembro.
 - Realizar un depósito.
 - Realizar una extraccion.
 - Acreditar intereses.
- Condinerar que algunas operaciones requieren comprobacion antes de ser realizadas
- Definir un programa principal que gestione dos cuentas distintas.
- Realizar (utilizando los metodos creados) la transferencia de una cuenta a la otra.

7. Representar mediante una clase la gestion de la informacion relativa a una persona. Dicha informacion consiste en:
- Apellido
 - Nombre
 - Fecha de nacimiento.
 - DNI
- Se deben definir los datos miembro necesarios y las funciones de carga y visualizacion de los mismos.
Definir ademas los metodos para:
Calcular la edad.
Mostrar los datos personales.
Mediante un mecanismo de herencia definir la clase alumno sabiendo que la unica informacion que se agrega es el curso que frequenta.
Determinar los metodos a definir para la clase derivada.
Determinar los metodos a redefinir si los hubiera.
8. Se desea obtener un sistema de representacion de figuras geometricas. Toda figura geometrica posee un centro (coordenada x y coordenada y) y posee ademas una formula para calcular su perimetro y su area.
Es posible definir tambien la razon superficie/perimetro.
Teniendo en cuenta estas cosideraciones obtenga un mecanismo de herencia que a partir de una clase abstracta FIG_GEO permita generar las clases derivadas para representar un punto, un círculo y un poligono regular.
Realice las pruebas de las clases creadas mediante un programa principal.
9. Contruir una clase que permita representar un arreglo unidimensional de enteros. Obtener las siguientes sobrecargas:
- Operador + para sumar dos arreglos.
 - Operador – para restar dos arreglos.
 - Operador * para calcular el producto escalar de 2 arreglos.
 - Operador [] para subindicar los elementos del arreglo.
 - Generalizar la clase mediante un template y crear arreglos de distintos tipos de datos.
- Utilizar las clases creadas en ejercicios anteriores.
10. Contruir la clase Matriz que permita representar un arreglo bidimensional de enteros. Obtener las siguientes sobrecargas:
- Operador + para sumar dos Matrices.
 - Operador – para restar dos Matrices.
 - Operador * para calcular el producto de 2 Matrices.
 - Operador () para subindicar cada elemento de la matriz.
 - Operador [] para subindicar los elementos de la diagonal principal
- Generalizar la clase mediante un template y crear arreglos de distintos tipos de datos. Utilizar las clases creadas en los ejercicios anteriores.

TRABAJO PRÁCTICO N° 4 - Archivos y Manejo de Excepciones

3. Objetivos de Aprendizaje

- a) Comprender el manejo de archivos y excepciones
- b) Construir programas en C++ utilizando archivos.
- c) Construir programas con manejo de excepciones.

2. Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad 8 de la programación de la asignatura. Los ejercicios 2 y 3 son de presentación obligatoria. Los demás son de presentación optativa.

3. Consignas a desarrollar en el trabajo práctico:

1. Crear un archivo de texto mediante un objeto de la clase ofstream, introducir en el mismo una tira de caracteres y posteriormente leerlo mediante un objeto de la clase ifstream.
2. Hacer nuevamente los ejercicios del 4 al 7 del práctico 4 de tal manera que la información resulte salvada en archivo y sea recuperada y posicionada en arreglos en memoria con cada nueva ejecución.
3. Obtener una clase que permita trabajar sobre un archivo de texto. La misma tendrá un dato miembro inicializado por el constructor que corresponde al nombre completo del archivo de texto a trabajar (path, nombre del archivo y extensión).
Determinar los datos y funciones miembro necesarios para proveer a la clase de las siguientes funcionalidades:
 - Lectura del texto.
 - Conversión a mayúsculas del texto.
 - Cálculo de la cantidad de caracteres del texto.
 - Cálculo de la cantidad de palabras del texto. (los espacios repetidos no deben ser considerados como palabras distintas).
 - Cálculo de la longitud promedio de las palabras del texto.
4. Obtener programas que haciendo uso de las excepciones (try catch) prevengan los siguientes errores de ejecución:
 - Memoria insuficiente al intentar la operación de new.
 - División por un denominador nulo.

TRABAJO PRÁCTICO EVALUABLE

Opción 1: GESTION DE PACIENTES EN UNA GUARDIA

Un hospital necesita un sistema para la gestión de los pacientes que acuden a la guardia. Los pacientes de esta área son atendidos por los profesionales en un cierto orden que depende del momento en que el paciente llega al hospital como así también de la gravedad de su afección. Cuando el paciente llega para ser atendido un enfermero lo evalúa y determina su nivel de prioridad asignándole uno de tres códigos: rojo (emergencia), amarillo (urgencia media) o verde (no urgente).

Para la implementación del sistema se pide definir las siguientes clases:

- **Clase paciente:** esta clase permitirá representar los pacientes que llegan al hospital. Para cada paciente se debe registrar su DNI y en el momento de ser evaluado se le asignará un código de paciente y un nivel de prioridad.
- **Clase guardia:** esta clase permitirá representar la dinámica de la guardia y para esto deberá contener un elenco de pacientes a ser atendidos. Se supone que la guardia tiene una capacidad máxima de pacientes en espera, si la capacidad se halla cubierta en un cierto momento, el paciente deberá ser derivado a otra institución y no ingresará en la lista de pacientes en espera.
Cada paciente nuevo debe ser evaluado para asignarle un código de paciente y una determinada prioridad. Para representar esta operatoria la clase tendrá un método "Ingreso" que recibirá como parámetros el DNI del paciente y el código de prioridad que se le ha asignado, generará el correspondiente código de paciente (único durante toda la jornada) y lo ubicará en la posición correspondiente dentro de la lista de espera. La ubicación en la lista de espera dependerá del código asignado (el más urgente debe estar antes) y también del momento de llegada, es decir a igual código de prioridad queda primero el paciente que llegó con anterioridad.
Los pacientes serán atendidos de acuerdo al orden en que se encuentran en esta lista de espera. Para simular la atención de un paciente se realizará un método "Atencion" que tomará el paciente que este en el primer lugar de la lista.
El método Ingreso agrega pacientes a la lista (hace crecer la lista), mientras que el método Atencion los extrae (achica la lista). Tener en cuenta que como se dijo anteriormente la capacidad del hospital está determinada (parámetro del constructor de la clase guardia) por lo que es posible que en algún momento no se pueda agregar pacientes a la lista hasta que alguno no sea atendido.

Para la operación de sistema se deberá instanciar un objeto de la clase guardia que iniciará con una lista de pacientes vacía pero con cierta capacidad. A medida que los pacientes lleguen a la guardia se irán ubicando en la lista y serán atendidos.

La clase guardia debe contar también con métodos que permitan obtener estadísticas sobre los pacientes que concurren al hospital. Será necesario en cada momento poder saber cuántos pacientes de cada prioridad hay en espera y cuantos pacientes de cada prioridad fueron atendidos en la jornada.

TRABAJO PRACTICO DE AULA F020

Por otro lado, el sistema deberá también generar un archivo de texto conteniendo los datos (DNI; código de paciente y código de prioridad) de todos los pacientes atendidos.

Se supone además que esta guardia trabaja solo en un cierto horario, el horario de cierre o de apertura no debe ser considerado en la definición de la misma, desde el programa principal se manejara esta situación haciendo que no se ingresen pacientes nuevos a partir de cierto momento. Aun cuando no se agreguen nuevos pacientes, no podrá culminarse la ejecución del programa hasta que todos los pacientes en espera hayan sido atendidos.

Opción 2: JUEGO DE LA OCA

El tablero del juego de la OCA es una secuencia de celdas numeradas de 1 a N. Cada celda puede asociarse una acción: adelantarse x cantidad de celdas o retroceder x cantidad de celdas. Hay dos celdas particulares que son la primera y la última, las cuales no tienen acciones asociadas. El objetivo del juego es llegar a la última celda (o pasarla) partiendo de la celda inicial.

Para esto el jugador tira un dado, de esa tirada sale un número entre 1 y 6 y la ficha del jugador deberá desplazarse ese número de celdas. Al llegar a la celda destino, según la acción asociada a la misma podrá avanzar más o deberá retroceder.

Para la implementación de juego se debe crear una clase con las siguientes características:

- **Clase OCA:** El constructor de la clase recibirá como parámetro el número de celdas de la misma y creará el tablero. Tener en cuenta que la primera y última celda no tienen acciones asociadas, pero todas las demás deben indicar el número de celdas que se avanza o retrocede cuando la ficha del jugador cae en ella. Podría también suceder que una de estas celdas no tenga acción asociada. La definición del tablero será aleatoria, de manera que en cada nueva jugada se tenga un tablero diferente.
Como mínimo deberá haber dos jugadores, el número de jugadores puede ser siempre fijo e igual a dos, pero el alumno que así lo desee puede plantear la solución con un número no fijo de jugadores (se debe especificar en el constructor) y en ese caso se deberá especificar en que orden irán jugado.
Una vez creado el tablero se podrá iniciar el juego, predisponer para ello un método que se encargue de posicionar a todos los jugadores sobre la primera celda.
Se debe proveer también un método jugar, que, según el orden establecido para los jugadores, realizara la jugada del mismo, es decir tirara el dado, de acuerdo al valor obtenido en la tirada, hará que el jugador se desplace a la celda correspondiente y de acuerdo a la acción encontrada en ella lo llevara la celda donde quedara finalmente posicionado.
Nunca un jugador puede quedar fuera del tablero, es decir antes de la celda de inicio o después de la celda de fin.
El juego deberá continuar hasta que alguno de los jugadores alcance la celda final y en ese caso este será el ganador del juego.
Durante el juego la clase deberá informar, del modo que se considere oportuno, la posición de cada jugador y el número de tiros que realizo. Generar la solución utilizando cuando será posible, métodos simples que resuelvan actividades puntuales.

Opción 3: PRESTAMOS BANCARIOS

Se requiere un sistema para el registro de los préstamos ofrecidos por una compañía financiera.

Los datos que se manejan para el préstamo son los siguientes:

- Número de Préstamo (numérico entero)
- Solicitante del préstamo: instancia de la clase Persona. La información por considerar es:
 - Nro. de Documento de identidad
 - Nombre
 - Apellido,
 - teléfono fijo
 - teléfono móvil.
- Valor del préstamo (numérico con decimales)
- Tasa de interés a aplicar (numérico entero en %)
- Numero de cuotas (numérico entero)
- Fecha de autorización del préstamo. (instancia de la clase fecha)
- Fecha de entrega del préstamo (Idem)
- Fechas de pago de las cuotas (arreglo de objetos tipo fecha)

Las reglas que debe respetar este proyecto son las siguientes:

- El número de préstamo siempre deberá ser un valor mayor que cero.
- El valor del préstamo siempre debe ser mayor a cero.
- Deben definirse métodos para la introducir valores en los datos miembros que se definan.
- La fecha de entrega del préstamo será siete días después de la fecha de autorización del préstamo.
- Las fechas de pago del préstamo se calculan, sumando 30 días a cada una a partir de la fecha de entrega del préstamo.
- Los cálculos de fechas de deben realizar utilizando operadores sobrecargados para la clase fecha.

Se debe generar como salida un archivo de texto con la información del préstamo: datos del solicitante, monto solicitado y la lista con el monto y vencimiento de todas las cuotas