

# LENGUAJES DE PROGRAMACIÓN

Trabajo Práctico - Septiembre de 2022

## Instrucciones

- El trabajo práctico debe realizarse de manera individual. No debe realizarse en grupo. Se penalizará cualquier uso compartido de las soluciones propuestas y de los códigos programados.
- El trabajo debe entregarse a través del curso virtual de la asignatura en la plataforma Alf.
- La fecha límite de entrega es el día 10 de septiembre.
- El alumno debe entregar un fichero comprimido, en formato zip o tar, que contenga:
  - Una *memoria* en la cual explique la solución a los ejercicios, incluyendo los listados documentados del código C++ desarrollado y una descripción detallada de las pruebas realizadas al código para comprobar que funciona correctamente. Este documento deberá estar en formato pdf.
  - Los ficheros del código fuente C++ solución a los ejercicios.

No deben entregarse ficheros ejecutables.

El nombre del fichero comprimido debe ser la concatenación de los apellidos y el nombre del alumno. Por ejemplo, GomezMartinLuisa.zip

## Criterios de evaluación

- Para que el trabajo pueda ser corregido, es imprescindible que el alumno entregue dentro del plazo establecido un fichero comprimido que contenga la memoria en formato pdf y el código fuente C++ de los ejercicios que haya realizado.
- El trabajo se compone de 4 ejercicios, cada uno de los cuales se valorará sobre 2.5 puntos.
- Para aprobar el trabajo es necesario que la nota total obtenida en los ejercicios sea mayor o igual que 5.
- Si el código solución de un ejercicio tiene errores de compilación o no tiene la funcionalidad pedida, dicho ejercicio se valorará con cero puntos.
- Si el código solución de un ejercicio compila sin errores y tiene la funcionalidad pedida, la puntuación en dicho ejercicio será al menos de 2 puntos.
- Se valorará positivamente la eficiencia y la adecuada documentación del código, así como la presentación y calidad de las explicaciones proporcionadas en la memoria.

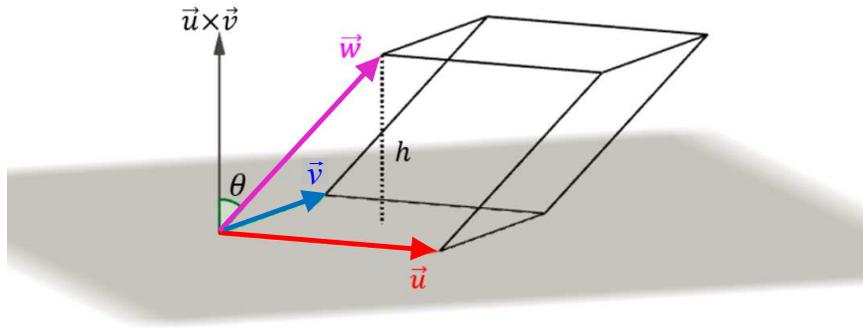
## Contenido de la memoria

Debe incluirse en la memoria, para cada uno de los ejercicios:

1. Listado del código fuente debidamente documentado.
2. Enumeración de las pruebas que usted ha realizado al código para comprobar que funciona correctamente.
3. Un ejemplo de ejecución del código, incluyendo capturas de pantalla en las que puedan verse las distintas fases en la ejecución del programa, tales como la entrada de los datos, la salida de los resultados, la escritura en la consola de mensajes de error, etc. según proceda en cada caso.

## Ejercicio 1

El volumen del paralelepípedo determinado por los vectores del espacio  $\vec{u}$ ,  $\vec{v}$  y  $\vec{w}$  es igual al área de la base determinada por los vectores  $\vec{u}$  y  $\vec{v}$ , multiplicada por la altura  $h$ .



Como puede observarse en la figura,  $\cos \theta = \frac{h}{|\vec{w}|}$ . El área de la base viene dada por el módulo del producto vectorial de los vectores  $\vec{u}$  y  $\vec{v}$ . Así pues, el volumen del paralelepípedo ( $V_p$ ) es igual al valor absoluto del producto mixto de los tres vectores  $\vec{u}$ ,  $\vec{v}$  y  $\vec{w}$ .

$$V_p = |h \cdot |\vec{u} \times \vec{v}|| = |\vec{w} \cdot (\vec{u} \times \vec{v})|$$

Escriba un programa en C++ que realice las acciones siguientes.

1. Escribir un mensaje en la consola solicitando al usuario que introduzca por consola las tres componentes cartesianas de los vectores del espacio  $\vec{u}$ ,  $\vec{v}$  y  $\vec{w}$ . Leer los valores introducidos por consola por el usuario, almacenándolos en sendos arrays de tres componentes **double** llamados **u**, **v** y **w**.
2. Calcular el valor absoluto de la altura,  $|h|$ , y escribirlo en la consola en formato fijo con 4 dígitos de precisión.
3. Calcular el volumen del paralelepípedo ( $V_p$ ) y escribirlo en la consola en formato científico, con 5 dígitos de precisión.
4. Terminar.

Escriba en la memoria el resultado obtenido al ejecutar su programa para los vectores  $\vec{u} = (2, -2, -1)$ ,  $\vec{v} = (1, 3, -2)$  y  $\vec{w} = (-1, 1, -1)$ .

## Ejercicio 2

Para poder realizar su actividad en el primer semestre, una empresa necesita disponer de un determinado número de ordenadores cada mes, según se indica en la Tabla 1.

**Tabla 1.** Número de ordenadores que necesita la empresa cada mes.

Mes	Ene.	Feb.	Mar.	Abr.	May.	Jun.
Ordenadores	9	5	7	8	10	5

La empresa decide alquilar los ordenadores. El precio de alquiler de un ordenador varía en función de la duración del alquiler:

- Si se alquila durante un mes, el precio son 400 euros.
- Si se alquila durante tres meses, el precio son 900 euros.

Debe decidirse cuántos ordenadores conviene alquilar al comienzo de cada mes por un periodo de un mes y cuántos ordenadores conviene alquilar al comienzo de cada mes por un periodo de tres meses. El objetivo es minimizar el coste total del alquiler, satisfaciendo exactamente las necesidades indicadas en la Tabla 1.

Definimos para ello las siguientes variables de tipo entero, que pueden tomar valor mayor o igual a cero:

$Ene_1, Ene_3$	Núm. ordenadores alquilados en enero durante 1 y 3 meses.
$Feb_1, Feb_3$	Núm. ordenadores alquilados en febrero durante 1 y 3 meses.
$Mar_1, Mar_3$	Núm. ordenadores alquilados en marzo durante 1 y 3 meses.
$Abr_1, Abr_3$	Núm. ordenadores alquilados en abril durante 1 y 3 meses.
$May_1,$	Núm. ordenadores alquilados en mayo durante 1 mes.
$Jun_1$	Núm. ordenadores alquilados en junio durante 1 mes.

El objetivo es encontrar un valor de las variables anteriores que minimice el coste total del alquiler:

$$\begin{aligned} \text{coste} = & 400 \cdot (Ene_1 + Feb_1 + Mar_1 + Abr_1 + May_1 + Jun_1) + \\ & 900 \cdot (Ene_3 + Feb_3 + Mar_3 + Abr_3) \end{aligned}$$

satisfaciéndose exactamente las necesidades mensuales de la empresa:

$$\begin{array}{rcl}
 Ene_1 + Ene_3 & = & 9 \text{ Ordenadores en enero} \\
 Feb_1 + Feb_3 + Ene_3 & = & 5 \text{ Ordenadores en febrero} \\
 Mar_1 + Mar_3 + Feb_3 + Ene_3 & = & 7 \text{ Ordenadores en marzo} \\
 Abr_1 + Abr_3 + Mar_3 + Feb_3 & = & 8 \text{ Ordenadores en abril} \\
 May_1 + Abr_3 + Mar_3 & = & 10 \text{ Ordenadores en mayo} \\
 Jun_1 + Abr_3 & = & 5 \text{ Ordenadores en junio}
 \end{array}$$

Escriba un programa en C++ que explore las posibles políticas de alquiler y que escriba en la consola todas aquellas (una o varias) que satisfagan exactamente las necesidades de la empresa y que tengan el menor coste.

El número de ordenadores que necesita la empresa cada mes y el coste del alquiler debe poder ser modificado de manera sencilla, por lo que deben declararse como constantes globales del programa de la forma siguiente:

```

const int NUM_ORD_ENE = 9;
const int NUM_ORD_FEB = 5;
const int NUM_ORD_MAR = 7;
const int NUM_ORD_ABR = 8;
const int NUM_ORD_MAY = 10;
const int NUM_ORD_JUN = 5;

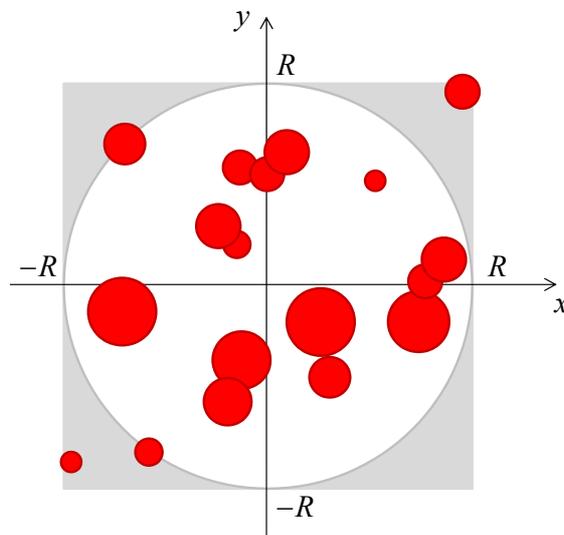
const double COSTE_UN_MES = 400;
const double COSTE_TRES_MESES = 900;

```

## Ejercicio 3

Consideremos una superficie cuadrada de lado  $L$ . Centrado sobre ella, se dispone un trozo de papel blanco de forma circular, con radio  $R = L/2$ . Cae un cierto número de gotas de tinta. Cada gota deja una mancha circular de color rojo, cuyo radio puede variar de una mancha a otra.

Para describir la ubicación de cada mancha emplearemos las coordenadas X-Y de su centro, en un sistema de coordenadas cartesiano cuyo origen está en el centro de la superficie cuadrada. En la figura se muestra un ejemplo.



En un fichero de texto llamado *manchas.txt* está almacenada la información acerca de la ubicación y tamaño de las manchas. Cada mancha es descrita en una línea del fichero. En las dos primeras columnas están las coordenadas  $X$  e  $Y$  del centro de la mancha, y en la tercera columna está escrito el radio.

Se desea estimar la proporción de la superficie del papel blanco que queda cubierta por las manchas de tinta. Para ello, se recubre la superficie cuadrada con una retícula compuesta por  $N \times N$  celdas cuadradas. La proporción de superficie del papel cubierta por las manchas de tinta vale aproximadamente  $N_{pm}/N_p$ , donde:

- $N_{pm}$  es el número de celdas de la retícula cuyo centro está simultáneamente contenido dentro del papel y dentro de al menos una mancha.
- $N_p$  es el número de celdas de la retícula cuyo centro está contenido dentro del papel.

Escriba un programa en C++ que realice las acciones siguientes.

1. Declarar una constante global de tipo entero llamada  $N$  y asignarle el valor 1000. Ésta determina el tamaño de la retícula.
2. Mediante un mensaje escrito en la consola, solicitar al usuario que introduzca por consola el radio del trozo circular de papel blanco. Leer el valor introducido por consola y almacenarlo en una variable de tipo **double** llamada  $R$ .
3. Abrir el fichero *manchas.txt* para lectura. Si se produce error al abrir el fichero, mostrar un mensaje en la consola indicándolo y terminar.  
Leer el fichero *manchas.txt*, almacenando su contenido en una estructura de datos. Puede suponer que el formato del fichero es correcto.
4. Estimar, de la manera descrita anteriormente, la proporción de la superficie del papel que está cubierta por las manchas de tinta, y escribir dicho valor en la consola.
5. Terminar.

Muestre en la consola el resultado obtenido al ejecutar su programa para el valor del radio  $R = 1$  y un fichero *manchas.txt* cuyo contenido es mostrado a continuación.

```
0.40  0.27  0.10
0.83  0.95  0.15
0.76 -0.34  0.20
0.12  0.09  0.10
-0.30 -0.66  0.21
0.71 -0.28  0.11
-0.04  0.93  0.01
0.97  0.04  0.21
-0.34  0.37  0.05
-0.68 -0.88  0.37
-0.62 -0.67  0.18
-0.01  0.26  0.19
-0.30 -0.66  0.11
0.41  0.95  0.05
-0.68 -0.20  0.07
0.07  0.01  0.16
-0.98  0.55  0.17
-0.18 -0.85  0.28
0.40  0.27  0.12
```

## Ejercicio 4

En un fichero de texto llamado *trayectorias.txt* se encuentran almacenadas las posiciones en el instante de tiempo  $t = 0$  y las velocidades de objetos puntuales que se mueven en el espacio con velocidad constante. Cada línea del fichero describe un objeto, siguiendo el formato siguiente:

<ID> <Xo> <Yo> <Zo> <Vx> <Vy> <Vz>

En la primera columna está escrita una palabra compuesta de letras y números, que identifica de manera unívoca al objeto. En las columnas dos a cuatro están escritas las coordenadas cartesianas del vector posición del objeto en el instante  $t = 0$ . En las columnas cinco a siete están escritas las coordenadas cartesianas del vector velocidad, el cual es constante en el tiempo.

Escriba un programa en C++ que lea los comandos introducidos por el usuario a través de la consola, realizando en cada caso la acción indicada en la Tabla 2.

**Tabla 2.** Formato de los comandos que debe aceptar el programa.

Comando	Acción
POS <ID> <TIME>	(1)(*)
VEL <ID>	(2)
DIS <ID1> <ID2> <TIME>	(3)(*)
QUIT	(4)

(\*) <TIME> representa un instante de tiempo, descrito mediante un número real.

- (1) Escribir en la consola las coordenadas cartesianas del vector posición del objeto con ese identificador (ID), en el instante de tiempo indicado.
  - (2) Escribir en la consola el módulo de la velocidad del objeto con ese ID.
  - (3) Escribir en la consola la distancia entre los dos objetos con los ID indicados, en el instante de tiempo indicado.
  - (4) Finalizar la ejecución del programa.
- (1)–(3) Si en el fichero no existe un objeto con el ID indicado, escribir en la consola el texto siguiente: *ID no encontrado*.

Si se produce error al abrir el fichero para lectura, el programa debe mostrar un mensaje en la consola indicándolo y terminar.

Al escribir el programa puede asumir que el contenido del fichero *trayectorias.txt* es correcto, en particular que el fichero no tiene errores de formato y no existen dos objetos con el mismo ID.

Si el texto introducido por consola por el usuario no se ajusta a ninguno de los comandos descritos en la Tabla 2, el programa debe mostrar un mensaje en la consola indicándolo.

A continuación se muestra, a modo de ejemplo, un posible contenido del fichero *trayectorias.txt*.

ORIG	0.00	0.00	0.00	0.00	0.00	0.00
R374	-0.14	12.48	2.39	12.34	3.22	-12.34
Uhytg	0.42	-0.34	12.99	12.55	-1.16	-33.12
U234	7.12	22.40	1.33	8.77	21.23	-14.13
NHGT4	12.99	13.26	-100.42	29.23	-12.34	-102.40
77YT6	-0.12	0.35	0.00	6.12	0.00	8.52

Muestre en la memoria el resultado obtenido al ejecutar su programa si el usuario introduce el texto siguiente por consola.

```

POS R374 10
POS R374a 10
POS R374a 10mal
PAS R374 10
VEL Uhytg
DIS NHGT4 R374 0.5
DIS NHGT4 R374m 0.5
QUIT

```