

Indicaciones específicas:

- Esta evaluación contiene 8 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un archivo fuente (.cpp) y un archivo cabecera (.h) con el número de la pregunta.
 1. p1.cpp, p1.h
 2. p2.cpp, p2.h
 3. p3.cpp, p3.h
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio. También puedes crear un .zip

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)
 - Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)
 - Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
 - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
 - Capacidad de aplicar conocimientos de ingeniería(nivel 2)
 - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	6	
2	7	
3	7	
Total:	20	

1. (6 points) Desarrollar la función **sumar_cruz** que permita crear una matriz cuadrada **n x n** a partir de una matriz cuadrada **n x n** donde los valores de la nueva matriz se obtengan a partir de la suma de todos los elementos alrededor que formen un cruz con el valor que ocupa la misma posición en la matriz original.

```
int** sumar_cruz(int** origen, int lado);
```

Se debe desarrollar un programa que lea un entero **n** que defina el tamaño del lado de la matriz, luego ingresar los valores de la matriz y utilizando la función **sumar_cruz** generar la nueva matriz, el programa debe mostrar la nueva matriz.

Algunos ejemplos de diálogo de este programa serían:

Ejemplo 1

Input

```
3
1 1 1
1 1 1
1 1 1
```

Output

```
3 4 3
4 5 4
3 4 3
```

Ejemplo 2

Input

```
4
2 2 2 2
2 2 2 2
2 2 2 2
2 2 2 2
```

Output

```
6 8 8 6
8 10 10 8
8 10 10 8
6 8 8 6
```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha implementado el cálculo de matrices con punteros para resolver el problema (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimizacion	El código es óptimo y eficiente (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).

2. (7 points) Desarrollar la función **obtener_palabras** que permita crear un vector de palabras de modo que a partir de un texto identifique cada palabra separados por un espacio, coma o punto.

```
vector<string> obtener_palabras(const string& origen);
```

Se debe desarrollar un programa que lea un texto y utilizando la función **obtener_palabras** genere el vector, el programa debe mostrar la palabra de menor tamaño en la primera línea e imprimir el resto de palabras del vector ordenadas alfabéticamente cada una en una línea. Si hubiese más de una palabra de tamaño menor se tomara la primera que aparezca en el texto.

Algunos ejemplos de diálogo de este programa serían:

Ejemplo 1

Input

```
este es un mensaje de prueba
```

Output

```
es
de
este
mensaje
prueba
un
```

Ejemplo 2

Input

```
hay golpes en la vida,tan fuertes.yo no se
```

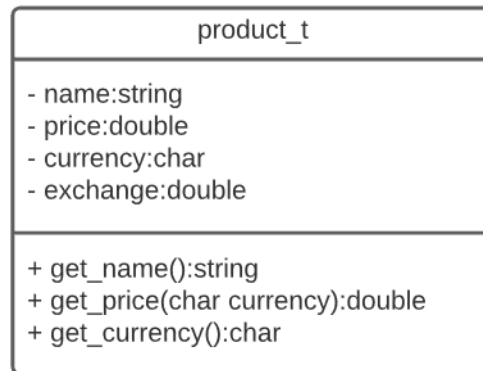
Output

```
en
fuertes
golpes
hay
la
no
se
tan
vida
yo
```

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha implementado vectores y sus métodos necesarios para resolver el problema (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación (3pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimizacion	El código es óptimo y eficiente (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).

3. (7 points) Desarrollar la clase **producto_t** que siga las características mostradas en el diagrama. donde el método **get_price** podra retornar el precio de acuerdo a la moneda que se eliga.



Se debe desarrollar un programa que lea un número **n** y que defina un vector de **n** productos lo que seran ingresados (nombre, moneda, tipo de cambio y precio), luego debe solicitar la moneda (S=Soles y D=Dolares) que servira para mostrar el total de precios.

Ejemplo 1

Input

```
3
Producto 1
S
3.6
100
Producto 2
D
3.0
150
Producto 3
S
3.5
200
S
```

Output

```
Producto 1 - 100
Producto 2 - 450
Producto 3 - 200
Total = 750
```

Ejemplo 2

Input

2
Camisa
D
3.6
200
Pantalon
S
3.5
105
D

Output

Camisa - 200
Pantalon - 30
Total = 230

La rúbrica para esta pregunta es:

Criterio	Logrado	Parcialmente Logrado	No Logrado
Código	Ha definido clases y objetos en forma correcta y lógica, así como sus métodos (3pts)	Existen algunos errores menores en la implementación (2pts)	El diseño y la implementación del código no son correctos (0pts).
Sintaxis	No existen errores sintácticos o de compilación (3pts)	Existen algunos errores sintácticos o de compilación. (1pts).	El código no compila (0pts).
Optimización	El código es óptimo y eficiente (1pts)	El código es optimizable en algunas partes (0.5pts).	El código es redundante y/o no es óptimo (0pts).