

Nombre y Apellido: N° Legajo:

Primer Parcial de Programación Imperativa

29/09/2023

	Ejercicio 1	Ejercicio 2	Ejercicio 3	Nota
Calificación	/3.5	/3.5	/3	

- ❖ **Condición mínima de aprobación: Sumar 5 (cinco) puntos.**
- ❖ **Los ejercicios que no se ajusten estrictamente al enunciado, no serán aceptados.**
- ❖ **No usar variables globales ni static.**
- ❖ **No es necesario escribir los #include**
- ❖ **Escribir en esta hoja Nombre, Apellido y Legajo**

Ejercicio 1

Dada una matriz de chars de COLS columnas -donde COLS es una constante previamente definida-, **verificar las filas que representan un palíndromo (capicúa)**. En cada celda puede haber letras, dígitos, símbolos, etc. Se debe considerar que las letras minúsculas no son equivalentes a las letras mayúsculas, por lo que "a1a" es **palíndromo** pero "A1a" no lo es.

Escribir una función elimina que reciba la matriz y la cantidad de filas de la misma, y **elimine de la matriz las filas que sean capicúas**. La función debe **retornar cuántas filas quedaron** en la matriz.

Ejemplo: Con COLS = 6, si la matriz es:

L	#	B	B	#	L
3	1	r	R	1	3
3	2	S	S	2	3
R	L	A	H	N	N
W	A	O	Y	U	T
Q	S	G	S	M	A

debe quedar de la siguiente forma y retornar **4**, donde lo que quede almacenado en las últimas dos filas no tiene importancia.

3	1	r	R	1	3
R	L	A	H	N	N
W	A	O	Y	U	T
Q	S	G	S	M	A

Ejercicio 2

Dada una matriz cuadrada de dimensión M -donde M es una constante simbólica par mayor a cero previamente definida (por ejemplo 2, 4, 6, etc.)-, se desea **calcular la suma de cada uno de sus bordes**. Se considera como un borde a un contorno de la matriz.

El primer borde estaría conformado por la primer y última fila junto a la primer y última columna, el segundo borde por la segunda y anteúltima fila junto a la segunda y anteúltima columna, y continuando así hasta llegar al centro de la matriz, formado por un cuadrado de 2×2 .

Ejemplo: Con $M = 6$, la siguiente matriz tiene 3 bordes, el primero formado por números 1, el segundo por números 2 y el tercero por números 3.

1	1	1	1	1	1
1	2	2	2	2	1
1	2	3	3	2	1
1	2	3	3	2	1
1	2	2	2	2	1
1	1	1	1	1	1

Escribir la función **bordes** que reciba:

- una matriz **mat** de enteros de dimensión M (no se debe validar que M sea par positivo, asumir que cumple con esa condición)
- un vector **sumas** de enteros no inicializado con al menos $M/2$ posiciones reservadas

La función debe dejar en `sumas[i]` la suma del borde i , para todo i en $[0, M/2)$.

En el ejemplo anterior el vector **sumas** debe quedar con los valores { 20, 24, 12 }, ya que el primer borde está formado por 20 celdas con el valor 1, el segundo borde por 12 celdas con el valor 2 y el tercer borde (el centro) por 4 celdas con el valor 3.

Si $M = 4$ y se recibe la siguiente matriz:

-2	2	-2	2
4	1	2	-4
0	1	2	0
-2	-2	2	2

en el vector **sumas** se deben dejar los valores { 0, 6 }

Si $M = 2$ y se recibe la siguiente matriz:

1	2
3	3

en el vector **sumas** se debe dejar el valor { 9 }

Ejercicio 3

Se dice que una palabra o frase es **anagrama** de otra palabra o frase **si ambas están compuestas por las mismas letras y con la misma cantidad de apariciones de cada letra**, sin tener en cuenta los blancos.

Por ejemplo "anagrama" es anagrama de "amar gana", pero no de "margana", ya que esta última tiene sólo 3 letras 'a' en vez de 4.

Escribir la función **anagrama** que **reciba dos strings** formados únicamente por letras del alfabeto inglés y espacios (no es necesario validarlo, se asume que ambos cumplen con esta condición) y **retorne 1 si ambos son anagramas o 0 si no lo son**.

Ejemplos de invocación:

```
char * s = "anagrama";

assert(anagrama(s, "amar gana"));
assert(anagrama("enrique", "quieren"));
assert(anagrama(s, s));
assert(anagrama("      ", ""));
assert(anagrama("amar gana", s));
assert(anagrama("", ""));
assert(anagrama("a gentleman", "elegant man"));

assert(anagrama("a", "") == 0);
assert(anagrama("mar", "amar") == 0);
```

En el ejemplo se muestran frases cortas, pero también podrían ser frases mucho más extensas