

Programación Orientada a Objetos 1

Práctica Calificada 2

Pregrado 2021-I

Profesor: Rubén Rivas Medina

Lab. 101

Indicaciones específicas:

- Esta evaluación contiene 9 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 solo archivo con el número de la pregunta. Por ejemplo:
 - 1. p1.cpp y p1.h
 - 2. p2.cpp y p2.h
 - 3. p3.cpp y p3.h
- Deberás subir estos archivos directamente a www.gradescope.com, o se puede crear un .zip que contenga todos ellos y subirlo.

Competencias y criterios de desempeño:

• Para los alumnos de la carrera de Ciencia de la Computación

Aplica conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa.(nivel 2)

Diseña, implementa y evalúa soluciones a problemas complejos de computación.(nivel 2)

Crea, selecciona, adapta y aplica técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones.(nivel 2)

• Para los alumnos de las carreras de Ingeniería

Capacidad para aplicar conocimientos de matemática. (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	7	
2	7	
3	6	
Total:	20	

1. (7 points) Escribir y diseñar una función ($sumar_numeros$) que retorne un vector de enteros y que permita leer un arreglo dinámico de enteros (n).

```
vector<int> sumar_numeros(int* arreglo,int n);
```

La función deberá retornar un vector que contenga solo 3 valores, el primero la suma de pares, la suma de impares y la suma de primos. Algunos ejemplos:

Ejemplo#1

Input Format

10 1 2 10 7 6 5 11 8 4 14

Output Format

44 24 25

Ejemplo #2

Input Format

6 1 2 10 7 6 5

Output Format

18 13 14

Ejemplo #3

Input Format

12 1 2 10 7 6 5 11 31 27 2 1 9

Output Format

20 92 58

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y	Elabora un al-	Elabora un al-	Elabora un al-	Elabora un al-
codificación	goritmo preciso,	goritmo preciso,	goritmo preciso,	goritmo preciso,
(4 pts)	definido y finito	definido y finito	definido y finito	definido y finito
	que da solución	que da solución	que da solución	que hace menos
	exacta a lo que	al menos al 80	al menos al 65	del 65 % de lo
	el enunciado re-	% de lo que	% de lo que	que el enunciado
	quiere. Utiliza	el enunciado re-	el enunciado	requiere. Utiliza
	arrays dinámicos	quiere. Utiliza	requiere. Utiliza	arrays dinámicos
	y hace un buen	arrays dinámicos	arrays dinámicos	y hace un buen
	uso de memoria:	y hace un buen	y hace un buen	uso de memoria:
	dimensiona, usa	uso de memoria:	uso de memoria:	dimensiona, usa
	y libera memoria	dimensiona, usa	dimensiona, usa	y libera memoria
	de manera ade-	y libera memoria	y libera memo-	de manera ade-
	cuada al codi-	de manera ade-	ria de manera	cuada al codi-
	ficar el algoritmo	cuada al codi-	adecuada al	ficar el algoritmo
	y lo hace con	ficar el algoritmo	codificar el al-	y lo hace con
	el 100% de pre-	y lo hace con	goritmo y lo	menos el 65% de
	cisión. $(4pts)$	al menos el 80%	hace con al	precisión. (0
		de precisión. (menos el 65%	pts)
		3pts)	de precisión.	
Sintaxis y	El algoritmo es	El algoritmo es	(2pts) El algoritmo es	El algoritmo
Sintaxis y legibilidad	El algoritmo es correcto, y es	El algoritmo es correcto, y es	correcto, y es	es incorrecto o
(1 pt)	codificado sin	codificado con	codificado con	es codificado
(1 pt)	errores de sin-	algunos errores	algunos errores	con errores de
	taxis. El nombre	de sintaxis,	de sintaxis,	sintaxis, que
	de las variables	pero que no	que afectan el	afectan el resul-
	y funciones son	afectan el resul-	resultado de	tado de manera
	descriptivas.	tado de manera	manera mínima,	significativa.
	(1 pts)	significativa.	o el nombre de	El nombre de
	(1 ***)	El nombre de	las variables y	las variables y
		las variables y	funciones no	funciones no
		funciones son	son descriptivas.	son descriptivas.
		descriptivas.	$(0.5 \mathrm{pts})$	(0pts)
		$(0.75 \mathrm{pts})$	/	
Optimización	El código es	El código es	El código es	El código es re-
de código	óptimo y efi-	óptimo en al	óptimo en al	dundante y/o no
(2 pt)	ciente (2pts)	menos el 80%	menos 65%	es $\acute{\mathrm{optimo}}(\mathtt{0pts})$
		(1.5pts)	(1pts)	

2. (7 points) Escribir y diseñar la función (esta_rodeado) que reciba una matriz cuadrada de lado (n) la cual almacenará los valores enteros 0s y 1s, en donde los 1s representarán los bordes y la función deberá detectar si los 0s no tocan (true) o tocan (false) los bordes.

```
bool esta_rodeado(int** matriz, int n);
```

Algunos ejemplos:

Ejemplo #1

Input Format

```
10
1 1 1 1 1 1 1 1
 1
   1
     0 0
          0 1
              1
                1
 1 1 0 0
          0 1 1
   1 0 0
         0 0
             1 1
 1 1 0 0 0 0 0 1 1
 1 0 0 0 0 0 1 1 1
 1 0 0 0 0 0 0 0 1
1 1 0 1 0 0 0 0 1 1
1 1 0 1 0
         0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```

Output Format

```
true
```

Ejemplo #2

Input Format

```
4
1 1 0 1
1 0 0 1
1 0 1 1
1 1 1 1
```

Output Format

```
false
```

Ejemplo #3

Input Format

Output Format

```
false
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y	Elabora un al-	Elabora un al-	Elabora un al-	Elabora un al-
codificación	goritmo preciso,	goritmo preciso,	goritmo preciso,	goritmo preciso,
(4 pts)	definido y finito	definido y finito	definido y finito	definido y finito
	que da solución	que da solución	que da solución	que hace menos
	exacta a lo que	al menos al 80	al menos al 65	del 65 % de lo
	el enunciado	% de lo que	% de lo que	que el enunci-
	requiere. Uti-	el enunciado	el enunciado	ado requiere.
	liza matrices	requiere. Uti-	requiere. Uti-	Utiliza matrices
	dinámicas y	liza matrices	liza matrices	dinámicas y
	hace un buen	dinámicas y	dinámicas y	hace un buen
	uso de memoria:	hace un buen	hace un buen	uso de memoria:
	dimensiona, usa	uso de memoria:	uso de memoria:	dimensiona, usa
	y libera memo-	dimensiona, usa	dimensiona, usa	y libera memo-
	ria de manera	y libera memo-	y libera memo-	ria de manera
	adecuada al	ria de manera	ria de manera	adecuada al
	codificar el algo-	adecuada al	adecuada al	codificar el
	ritmo y lo hace	codificar el al-	codificar el al-	algoritmo y
	con el 100%	goritmo y lo	goritmo y lo	lo hace con
	de precisión.	hace con al	hace con al	menos el 65%
	(4pts)	menos el 80%	menos el 65%	de precisión. (0
		de precisión. (de precisión.	$ ext{pts})$
G: .	T21 1	3pts)	(2pts)	T)
Sintaxis y	El algoritmo es	El algoritmo es	El algoritmo es	El algoritmo
legibilidad	correcto, y es	correcto, y es	correcto, y es	es incorrecto o
(1 pt)	codificado sin	codificado con	codificado con	es codificado
	errores de sin- taxis. El nombre	algunos errores	algunos errores	con errores de
	de las variables	de sintaxis,	de sintaxis,	sintaxis, que afectan el resul-
	y funciones son	pero que no afectan el resul-	que afectan el resultado de	tado de manera
	descriptivas.	tado de manera	manera mínima,	significativa.
	(1pts)	significativa.	o el nombre de	El nombre de
	(TPus)	El nombre de	las variables y	las variables y
		las variables y	funciones no	funciones no
		funciones son	son descriptivas.	son descriptivas.
		descriptivas.	(0.5pts)	(0pts)
		(0.75pts)	(3.3 p 3.3)	(°P°°)
Optimización	El código es	El código es	El código es	El código es re-
de código	óptimo y efi-	óptimo en al	óptimo en al	dundante y/o no
(2 pt)	ciente (2pts)	menos el 80%	menos 65%	es óptimo(0pts)
		(1.5 pts)	(1pts)	

3. (6 points) Escribir la función (obtener_repetidos) que tenga como parámetro un vector de numeros enteros cuyo rango es desde 1 hasta 60 y un parámetro (repeat) un número entero que cuente las repeticiones la función debe generar un vector ordenado con todos aquellos números que se repitan las veces que se solicitan.

```
vector<int> obtener_repetidos(vector<int>& vec, int repeat);
```

Algunos ejemplos: Ejemplo #1

Input Format

```
9
1 2 5 7 8 5 4 8 8
2
```

Output Format

5

Ejemplo #2

Input Format

```
20
1 3 6 5 4 12 31 20 11 2 6 3 4 5 11 59 21 4 50 11
3
```

Output Format

```
4 11
```

Ejemplo #3

Input Format

```
16
1 2 10 7 6 5 11 15 10 3 1 9 13 20 14 4
1
```

Output Format

```
2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 9 \ 11 \ 13 \ 14 \ 15 \ 20
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y	Elabora un al-	Elabora un al-	Elabora un al-	Elabora un al-
codificación	goritmo preciso,	goritmo preciso,	goritmo preciso,	goritmo preciso,
(3 pts)	definido y finito	definido y finito	definido y finito	definido y finito
	que da solución	que da solución	que da solución	que hace menos
	exacta a lo que	al menos al 80	al menos al 65	del 65 % de lo
	el enunciado	% de lo que	% de lo que	que el enunciado
	requiere. Uti-	el enunciado	el enunciado	requiere. Uti-
	liza vectores y	requiere. Uti-	requiere. Uti-	liza vectores y
	sus métodos al	liza vectores y	liza vectores	sus métodos
	codificar el algo-	sus métodos	al codificar el	al codificar el
	ritmo y lo hace	al codificar el	algoritmo y	algoritmo y
	con el $100%$	algoritmo y	lo hace con al	lo hace con
	de precisión.	lo hace con al	menos el 65%	menos el 65%
	(4pts)	menos el 80%	de precisión.	de precisión. (0
		de precisión. ((1pts)	pts)
		2pts)		
Sintaxis y	El algoritmo es	El algoritmo es	El algoritmo es	El algoritmo
legibilidad	correcto, y es	correcto, y es	correcto, y es	es incorrecto o
(1 pt)	codificado sin	codificado con	codificado con	es codificado
	errores de sin-	algunos errores	algunos errores	con errores de
	taxis. El nombre	de sintaxis,	de sintaxis,	sintaxis, que
	de las variables	pero que no	que afectan el	afectan el resul-
	y funciones son	afectan el resul-	resultado de	tado de manera
	descriptivas.	tado de manera	manera mínima,	significativa.
	(1pts)	significativa.	o el nombre de	El nombre de
		El nombre de	las variables y	las variables y
		las variables y	funciones no	funciones no
		funciones son	son descriptivas.	son descriptivas.
		descriptivas.	$(0.5 \mathrm{pts})$	(0pts)
		(0.75 pts)		
Optimización	9	El código es	El código es	El código es re-
de código	óptimo y efi-	óptimo en al	óptimo en al	dundante y/o no
(2 pt)	ciente (2pts)	menos el 80%	menos 65%	es $optimo(0pts)$
		$(1.5 \mathrm{pts})$	(1pts)	