



CS1102 Programación Orientada a Objetos

Punteros, asignación dinámica de memoria y vectores

Pregrado 2021-1

Ciencia de la Computación

Laboratorio

1. Desarrolle e implemente un programa que solicite el tamaño de una matriz cuadrada, y realice lo siguiente:

- Generar dos matrices cuadradas **dinámicas** con valores aleatorios enteros entre 1 y 5
- Imprimir ambas matrices
- Generar e imprimir una tercera matriz, también cuadrada que tenga el número 1 en las posiciones en que los valores de las dos primeras matrices coincidan tanto en valor como en posición y cero si no coinciden.
- Es necesario que utilice punteros y asignación dinámica de memoria
- No olvide liberar la memoria la final del programa.

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Orden de la matriz cuadrada: 4

Matriz 1:

    4    2    3    1
    4    1    2    3
    5    2    3    3
    1    5    4    2

Matriz 2:

    1    2    3    2
    2    4    3    5
    3    1    3    4
    3    1    5    3

Matriz de Coincidencias:

    0    1    1    0
```

0	0	0	0
0	0	1	0
0	0	0	0

Algunos ejemplos de diálogo de este programa serían:

Listing 2: Ejemplo 2

```
Orden de la matriz cuadrada: 6
```

```
Matriz 1:
```

4	2	3	1	4	1
2	3	5	2	3	3
1	5	4	2	1	2
3	2	2	4	3	5
3	1	3	4	3	1
5	3	3	4	5	3

```
Matriz 2:
```

4	2	2	3	5	4
2	5	5	3	4	5
1	1	4	2	2	1
2	4	3	1	2	2
1	1	5	3	2	1
2	5	4	3	5	1

```
Matriz de Coincidencias:
```

1	1	0	0	0	0
1	0	1	0	0	0
1	0	1	1	0	0
0	0	0	0	0	0
0	1	0	0	0	1
0	0	0	0	1	0

Algunos ejemplos de diálogo de este programa serían:

Listing 3: Ejemplo 3

```
Orden de la matriz cuadrada: 3
```

```
Matriz 1:
```

4	2	3
1	4	1

2	3	5
Matriz 2:		
2	3	3
1	5	4
2	1	2
Matriz de Coincidencias:		
0	0	1
1	0	0
1	0	0

2. Desarrolle e implemente un programa que solicite el tamaño de una matriz cuadrada, y realice lo siguiente:
- Generar dos matrices cuadradas **dinámicas** con valores aleatorios enteros entre 1 y 5
 - Imprimir ambas matrices
 - Generar e imprimir una tercera matriz, también cuadrada. En cuyos bordes contenga la suma de cada elemento de las dos matrices anteriores. Los elementos que no estén ubicados en los bordes no se suman y en la tercera matriz tendrán el valor de cero.
 - Es necesario que utilice punteros y asignación dinámica de memoria
 - No olvide liberar la memoria la final del programa.

Algunos ejemplos de diálogo de este programa serían:

Listing 4: Ejemplo 1

Orden de la matriz cuadrada: 6					
Matriz 1:					
4	2	3	1	4	1
2	3	5	2	3	3
1	5	4	2	1	2
3	2	2	4	3	5
3	1	3	4	3	1
5	3	3	4	5	3
Matriz 2:					
4	2	2	3	5	4

```
2    5    5    3    4    5
1    1    4    2    2    1
2    4    3    1    2    2
1    1    5    3    2    1
2    5    4    3    5    1
```

Matriz Resultado:

```
8    4    5    4    9    5
4    0    0    0    0    8
2    0    0    0    0    3
5    0    0    0    0    7
4    0    0    0    0    2
7    8    7    7    10   4
```

Listing 5: Ejemplo 1

Orden de la matriz cuadrada: 8

Matriz 1:

```
4    2    3    1    4    1    2    3
5    2    3    3    1    5    4    2
1    2    3    2    2    4    3    5
3    1    3    4    3    1    5    3
3    4    5    3    4    2    2    3
5    4    2    5    5    3    4    5
1    1    4    2    2    1    2    4
3    1    2    2    1    1    5    3
```

Matriz 2:

```
2    1    2    5    4    3    5    1
3    1    5    3    5    5    4    1
3    4    2    4    4    5    4    2
5    5    3    1    2    4    5    3
2    2    5    5    1    1    5    4
3    2    3    3    3    3    3    2
2    1    2    2    1    5    5    5
1    5    2    3    3    2    2    1
```

Matriz Resultado:

```
6    3    5    6    8    4    7    4
8    0    0    0    0    0    0    3
```

4	0	0	0	0	0	0	7
8	0	0	0	0	0	0	6
5	0	0	0	0	0	0	7
8	0	0	0	0	0	0	7
3	0	0	0	0	0	0	9
4	6	4	5	4	3	7	4

Listing 6: Ejemplo 1

Orden de la matriz cuadrada: 4			
Matriz 1:			
4	2	3	1
4	1	2	3
5	2	3	3
1	5	4	2
Matriz 2:			
1	2	3	2
2	4	3	5
3	1	3	4
3	1	5	3
Matriz Resultado:			
5	4	6	3
6	0	0	8
8	0	0	7
4	6	9	5

3. Diseñe e implemente un programa que permita crear un **vector** de "n" números enteros generados al azar, cuyo rango de valores estén entre 1 y 999, y luego realice lo siguiente:
- Imprimir el vector
 - Formar e imprimir un nuevo vector, en donde figuren solamente aquellos números del primer vector que tengan algún dígito igual a 4.
 - Para asignar el puntaje a esta pregunta, es necesario que se forme en nuevo vector.

Algunos ejemplos de diálogo de este programa serían:

Listing 7: Ejemplo 1

Numero de elementos: 23

```

Vector con valores al azar :
  479  665  154  269  501  998  992  904  763  254  591  869
      843  683  708  410   88  352  566  497  252  486  565

Vector nuevo, con valores de n meros que al menos tengan el
  4:
  479  154  904  254  843  410  497  486

```

Listing 8: Ejemplo 2

```

Numero de elementos: 10

Vector con valores al azar :
  479  665  154  269  501  998  992  904  763  254

Vector nuevo, con valores de n meros que al menos tengan el
  4:
  479  154  904  254

```

4. Desarrollar un programa en C++ usando la clase vector. El programa debe rotar una cadena a la derecha o izquierda. Debe ingresar 2 parámetros desde teclado, CADENA y NUMERO.

- CADENA, es un string que se debe copiar a un vector.
- NUMERO, es un entero que indica el número de posiciones que se rotan
 Sí es positivo rota a la derecha
 Sí es negativo rota a la izquierda

Algunos ejemplos de diálogo de este programa serían:

Listing 9: Ejemplo 1

```

CADENA : Utec
NUMERO:  1
RESULTADO: cUTe

```

Listing 10: Ejemplo 2

```

CADENA : Utec
NUMERO:  -1
RESULTADO: tecU

```

5. Ingresar desde teclado el valor para N. Generar N valores aleatorios (usar obligatorio punteros). El rango aleatorio es entre 1 y 10.
- Contar el número que se repite más veces, (pueden ser varios)

Listing 11: Ejemplo 1

```
N: 6
1
3
2
3
3
8

Respuesta: Numero que mas se repite
3
```

Listing 12: Ejemplo 2

```
N: 10

7
6
3
2
3
7
2
8
7
2

Respuesta: Numero que mas se repite
7
2
```

6. Ingresar desde teclado el valor de N, con el cual se genera una matriz cuadrada **dinámicas** de NxN. Los valores generados son aleatorios con el rango de 1-9.
- Ingresar desde teclado el valor para C, en cual indica el número de columna.
 - Ingresar desde teclado el valor para OP, el cual indica la operación
 - OP= S, realiza la SUMA de los valores
 - OP= M, realiza la MULTIPLICACIÓN de los valores

Algunos ejemplos de diálogo de este programa serían:

Listing 13: Ejemplo 1

```
n: 4
c: 2
op= S
```

```
1 3 4 2
7 3 2 2
3 4 7 1
1 2 2 3
```

Respuesta:
3+3+4+2=12

Listing 14: Ejemplo 2

```
n: 4
c: 1
op= M
1 3 4 2
7 3 2 2
3 4 7 1
1 2 2 3
```

Respuesta:
1x7x2x1=14

7. Implemente una función que tome 3 parámetros de entrada y 1 parámetro de salida, y genere números en un vector o array, según la opción pasada, de la siguiente forma:

- El primer parámetro es un número entero que marca el número de inicio.
- El segundo parámetro es un número entero que marca el número final.
- El tercer parámetro es una caracter que define una opción:
 - p genera números pares,
 - i genera números impares,
 - e genera números múltiplos de 3.
- El cuarto parámetro es un vector o array donde se escriben los números generados

Luego implemente un programa que solicite ingresar esos 3 parámetros y llame a la función, e imprime el vector o array resultante.

Algunos ejemplos de diálogo de este programa serían:

Listing 15: Ejemplo 1

```
Numero 1: 4
Numero 2: 10
Opcion: p
Salida: 4 6 8 10
```


Listing 16: Ejemplo 2

```
Numero 1: 5
Numero 2: 12
Opcion: i
Salida: 5 7 9 11
```

Listing 17: Ejemplo 2

```
Numero 1: 5
Numero 2: 12
Opcion: e
Salida: 6 9 12
```

8. Implemente una función que recibe dos vectores, y un número entero, y la función debe intercambiar los valores de los dos vectores desde la posición indicada por el número del parámetro hasta el final.

Luego implemente un programa que solicite ingresar valores para el vector 1, vector 2 y la posición, debe llamar a la función con esos parametros y debe imprimir los vectores resultantes.

Vector 1 y Vector 2 pueden tener diferentes tamaños.

Algunos ejemplos de diálogo de este programa serían:

Listing 18: Ejemplo 1

```
tam 1: 4
vector 1: 23 45 56 78
tam 2: 4
vector 2: 29 18 10 23
posicion: 2
salida vector 1: 23 45 10 23
salida vector 2: 29 18 56 78
```

Listing 19: Ejemplo 2

```
tam 1: 6
vector 1: 12 45 78 23 45 88
tam 2: 7
vector 2: 12 56 86 95 74 84 12
posicion: 5
salida vector 1: 12 45 78 23 45 84 12
salida vector 2: 12 56 86 95 74 88
```

9. Implemente un programa que pide al usuario 3 palabras, y genere una matriz dinamica de caracteres, considere el tamaño de la matriz con tal que permita almacenar las 3

palabras, y dentro de esa matriz deben estar esas 3 palabras una en horizontal, otra en vertical y otra en diagonal, tipo pupiletras.

Algunos ejemplos de diálogo de este programa serían:

Listing 20: Ejemplo 1

```
palabra 1: hola
palabra 2: como
palabra 3: estan
f c e e t d j
h o l a d d i
j m e d f e o
i o s s u e u
r y u j t d o
u r u f r a n
e r t y u e n
```

10. 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
```

11. 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
```

12. Desarrollar la función **reducir_matriz** que permita crear una matriz cuadrada **m x m** a partir de una matriz cuadrada **n x n** donde los valores de la nueva matriz se obtengan removiendo la fila y columna mencionada.

```
int** reducir_matriz(int** origen, int lado, int
    linea_remove);
```

Se debe desarrollar un programa que lea un entero **n** que sirva para definir el tamaño del lado de la matriz, luego ingresar los valores de la matriz e ingresar un número que servirá para determinar la fila y columna que debe ser borrada, luego con esta información y utilizando la función **reducir_matriz** generar la nueva matriz, el programa debe mostrar la nueva matriz.

Algunos ejemplos de diálogo de este programa serían:

Ejemplo 1

Input

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

Output

```
1 3
7 9
```

Ejemplo 2

Input

```
4
2 2 3 2
2 2 4 2
3 4 5 6
```

```
2 2 6 2
2
```

Output

```
2 2 2
2 2 2
2 2 2
```