Programación II
Ejercicios
Arreglos y matrices
Pregrado2023-II
Profesor: Edson Mendiolaza Cornejo

1. Elabore un programa que genere números aleatorios entre 0 y 100 en un arreglo dinámico de tamaño definido por el usuario. Construya la función cambiaValor que permita al usuario modificar el valor de un elemento del array dinámico. Esta función recibirá como parámetros los siguientes valores:

   - Puntero al arreglo (ptrArreglo): Permite modificar el valor solicitado
   - Elementos (elementos): El número de elementos del arreglo.
   - Nuevo valor (nuevo): Es un número entero ingresado por el usuario.
   - Indice (indice): Es un número entero, la posición en el arreglo del elemento a ser modificado.

   La función no retorna ningún valor. El programa principal incluirá un ciclo repetitivo que permitirá al usuario seguir modificando valores. El programa termina cuando el usuario ingresa -1 como posición.

   Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Ingrese el numero de elementos:5

70 72 10 16 83

Ingrese posicion      a cambiar (-1 para finalizar):0

Ingrese el nuevo      valor:100

100 72 10 16 83

Ingrese posicion      a cambiar (-1 para finalizar):-1
Process finished      with exit code 0
```

Listing 2: Ejemplo 2

```
Ingrese el numero de elementos:6

36 91 85 32 4 28

Ingrese posicion a cambiar (-1 para finalizar):3

Ingrese el nuevo valor:400

36 91 85 400 4 28

Ingrese posicion a cambiar (-1 para finalizar):1

Ingrese el nuevo valor:700
```

```
36 700  85 400  4  28

Ingrese posicion a cambiar (-1 para finalizar):-1

Process finished with exit code 0
```

Listing 3: Ejemplo 3

```
Ingrese el numero de elementos:5

15 22 44 96 79

Ingrese posicion a cambiar (-1 para finalizar):-1

Process finished with exit code 0
```

Listing 4: Ejemplo 4

```
Ingrese el numero de elementos:4

16 43  78  63

Ingrese posicion      a cambiar (-1 para finalizar):1

Ingrese el nuevo      valor:555

16 555  78  63

Ingrese posicion      a cambiar (-1 para finalizar):-1

Process finished      with exit code 0
```

2. Utilizando funciones y punteros implementar el código necesario para recibir un array de enteros y poder identificar el menor y mayor elemento.

Input

- La primera línea es un número entero que determina la longitud del array.
- La segunda línea son números enteros que representan los elementos del array separados por espacios.

Ouput

- Emprimir el menor elemento, el mayor elemento y separados por un espacio.

Algunos ejemplos de este programa serían:

Listing 5: Ejemplo 1

```
Ingresar las elementos:
2 3 5 6 4 6 7 9 7
Output:
2 9
```

Listing 6: Ejemplo 2

```
Ingresar las elementos:
15 2 5 8 9 4 58 1 9
Output:
1 58
```

Listing 7: Ejemplo 3

```
Ingresar las elementos:
5 87 63 1 7 5 9 7 3 5
Output:
3 87
```

3. Crear una función **letras dobles** que tome como parametro una palabra y que retorne true (verdadero) si la palabra tiene 2 letras consecutivas iguales.

   Algunos ejemplos de diálogo de este programa serían:

Listing 8: Ejemplo 1

```
    Ingrese una palabra: look
    La palabra tiene letras dobles
```

Listing 9: Ejemplo 2

```
    Ingrese una palabra: coco
    La palabra NO tiene letras dobles
```

Listing 10: Ejemplo 2

```
    Ingrese una palabra: fresa
    La palabra NO tiene letras dobles
```

Listing 11: Ejemplo 3

```
    Ingrese una palabra: correr
    La palabra tiene letras dobles
```

4. Crear un programa en c++ que dado dos números ingresados por el usuario A y B donde A tiene que ser menor que B. Se pide mostrar todos los números primos existentes desde A hasta B.

   - Validar que A sea menor que B.
   - Realizar la asignación en la memoria de forma dinámica.
   - Compilación: *g++ -c main.cpp funciones.h implementacion.cpp*

   Algunos ejemplos de diálogo de este programa serían:

Listing 12: Ejemplo 1

```
Ingrese  A: 9
Ingrese  B: 1
Ingrese  A: 1
Ingrese  B: 20
Los n meros primos desde 1 hasta 20 son:
2, 3, 5, 7, 11, 13, 17, 19,
```

Listing 13: Ejemplo 2

```
Ingrese A: 10
Ingrese B: 20
Los n meros primos desde 10 hasta 20 son:
11, 13, 17, 19,
```

Listing 14: Ejemplo 3

```
Ingrese A: 15
Ingrese B: 16
No existen numeros primos desde 15 hasta 16
```

Listing 15: Ejemplo 4

```
Los n meros primos desde 100 hasta 109 son:
101, 103, 107, 109,
```

5. Escribe un programa que use e implemente la función great. Esta función deberá recibir un arreglo de caracteres y deberá retornar true si es posible armar la palabra CS-UTEC. En caso contrario retornará false.

   Algunos ejemplos de diálogo de este programa serían:

Listing 18: Ejemplo 1

```
Ingrese caracteres hasta presionar "espacio".
S
U
E
T
-
C
C
Z

true
```

Listing 19: Ejemplo 2

```
Ingrese caracteres hasta presionar "espacio".
A
M
E
P
-
S
D
H

false
```

6. Escribir un programa que valide la CLAVE de un alumno. Debe usar 3 FUNCIONES para las siguientesvalidaciones:

   - M´ınimo 8 caracteres
   - Debe contener 2 d´ıgitos
   - Debe contener una minúscula

   Si cumple todas las condiciones debe indicar el mensaje “CLAVE correcta” en caso contrario “CLAVEincorrecta”

   Algunos ejemplos de diálogo de este programa serían:

Listing 20: Ejemplo 1

```
clave: Universidad
Respuesta:  CLAVE incorrecta
```

Listing 21: Ejemplo 2

```
clave: Universidad34
Respuesta:  CLAVE correcta
```

7. Elabore un programa que solicite al usuario el ingreso de un número entero N. Luego se generan dos arrays dinámicos de N elementos de tipo entero generados aleatoriamente. Se imprimen ambos arrays y luego se intercambian los valores de ambos arrays. Se sugiere el uso de un array dinámico temporal.

   Algunos ejemplos de diálogo de este programa serían:

Listing 22: Ejemplo 1

```
Ingrese el numero de elementos:5

Arreglo:  97 40 75 27 34
Arreglo:    18 8 21 97 43

Luego del   intercambio:

Arreglo:    18 8 21 97 43
Arreglo:    97 40 75 27 34

Process finished with exit code 0
```

Listing 23: Ejemplo 2

```
Ingrese el numero de elementos:3

Arreglo:    98 61 8
Arreglo:    25 6 75

Luego del   intercambio:

Arreglo:    25 6 75
Arreglo:    98 61 8

Process finished with exit code 0
```

Listing 24: Ejemplo 3

```
Ingrese el numero de elementos:7

Arreglo:    77 78 64 89    80   12  53
Arreglo:    61 31 79 38    11   51  88

Luego del   intercambio:

Arreglo:    61 31 79 38    11   51  88
Arreglo:    77 78 64 89    80   12  53

Process finished with exit code 0
```

8. Sean *A* y *B* dos conjuntos. La diferencia de *A* y *B* se denota por *A* − *B* y es el conjunto de los elementos de *A* que no están en *B*, como se puede observar en la figura 1.

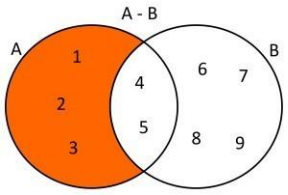


Figure 1: Imagen referencial

En la figura 1 se observa que solo interesan los elementos del conjunto A que no estén en B, siendo $A - B = \{1, 2, 3\}$.

De lo anterior asuma que *A* y *B* tienen 10 elementos de tipo entero respectivamente. Se requiere elaborar un programa que permita guardar los elementos de *A* − *B* en un array dinámico y luego imprimirlo.

Input

- Elementos del conjunto A y B son enteros positivos mayores a 0.

Ouput

- Imprimir los elementos del array dinámico que es el resultado A-B.

Algunos ejemplos de este programa serían:

Listing 25: Ejemplo 1

```
Ingresar las elementos:
13 14 15 7 14 3 1 18 20 4
1 10 15 6 18 7 15 19 11 7
Output: 13 14 14 3 20 4
```

Listing 26: Ejemplo 2

```
Ingresar las elementos:
15 15 6 10 3 20 7 3 4 12
19 6 18 19 16 15 15 4 13 19
Output: 10 3 20 7 3 12
```

Listing 27: Ejemplo 2

```
Ingresar las elementos:

 9 11 13  3  6 15  17  2 2 5
 10 14 3  1  13 2  12  15 12 10
Output:  9  11 6  17  5
```

9. Imaginalandia le declara la guerra al Perú, el general de dicho país decide primero enviar una unidad de reconocimiento a la zona fronteriza. Para lo cual debe dar indicaciones al comándate de división sobre la número de tanques y la cantidad de soldados en cada tanque. Toda esta información es transmitida en un solo número encriptado. Afortunadamente, la inteligencia peruana halló el método de descifrar dicho número y consiste en lo siguiente:

   El número de tanques corresponde a la mitad de dígitos del número encriptado. La cantidad de soldados en cada tanque es la suma de los dos dígitos que aparecen en posición simétrica. Si el número de d´ıgitos es impar, se ignora el digito central.

```
#include <iostream>
using namespace std;
int* descomponerDigitos(long int numero,int &n)
{
        //implemente usted
}

int* llenarTanques(int* digitos,int  n, int &m)
{
        //implemente usted
```

```
}

void mostrarTanques(int* tanques, int m)
{
        //implemente usted
}
int main() {
        long int numero;
        cout<<"Numero.encriptado:";  cin>>numero;
        int n = 0, m = 0;
        int *digitos = descomponerDigitos(numero, n);
        int *tanques = llenarTanques(digitos, n, m);
        mostrarTanques(tanques, m);
        return 0;
}
```

Implemente usted dos de tres de las funciones.

Algunos ejemplos de diálogo de este programa serían:

```
Numero encriptado: 31524
-----
Numero de tanques: 2
Tanque 1: 7 soldados
Tanque 2: 3 soldados
```

```
Numero encriptado: 245834
-----
Numero de tanques: 3
Tanque 1: 6 soldados
Tanque 2: 7 soldados
Tanque 3: 13 soldados
```

10. Escribe un programa que reciba como parametro una palabra y que genere un arreglo dinamico deenteros que almacene los codigos ascii de todas las vocales en el orden en que aparecen en la palabrae imprimir, letra por letra en cada linea.

Algunos ejemplos de diálogo de este programa serían:

Listing 28: Ejemplo 1

```
Ingrese una palabra: correr
los codigos ascii de las vocales son:
111
101
```

Listing 29: Ejemplo 1

```
Ingrese una palabra: doris
los codigos ascii de las vocales son:
111
105
```

Listing 30: Ejemplo 1

```
Ingrese una palabra: ahora
los codigos ascii de las vocales son:
97
111
97
```

Listing 31: Ejemplo 1

```
Ingrese una palabra: nnn
No tiene vocales
```

11. Crear un programa que genere 50 números aleatorios desde 100 hasta 500 usando asignación dinámica. Adicionalmente, se pide generar los siguientes reportes a partir del siguiente Menu:

MENU

1. Mostrar los números pares existentes desde 200 hasta 300.
2. Mostrar los números multiplos de 3 existentes desde 100 hasta 200.
3. Mostrar los números multiplos de 7 existentes en los 50 números.
4. Salir del programa. Tomar en

cuenta lo siguiente:

- Realizar la asignación en la memoria de forma dinámica.
- Compilación: *g++ -c main.cpp funciones.h implementacion.cpp*

Algunos ejemplos de diálogo de este programa serían:

Listing 32: Ejemplo 1

```
                    MENU
1.- Mostrar los n meros pares existentes desde 200 hasta 300.
2.- Mostrar los n meros multiplos de 3 existentes desde 100 hasta
   200.
3.- Mostrar los n meros multiplos de 7 existentes en los 50 numeros.
4.- Salir del programa

Ingrese opcion: 1
146, 126, 460, 412, 394, 284, 490, 238, 372, 156, 324, 188, 272, 352,
   346, 232, 206, 290, 216, 500, 184, 400, 368, 462, 418, 108,
```

Listing 33: Ejemplo 2

```
                    MENU
1.- Mostrar los n meros pares existentes desde 200 hasta 300.
2.- Mostrar los n meros multiplos de 3 existentes desde 100 hasta
   200.
3.- Mostrar los n meros multiplos de 7 existentes en los 50 numeros.
4.- Salir del programa

Ingrese opcion: 2
126, 372, 156, 324, 405, 453, 429, 465, 216, 462, 189, 105, 108,
```

Listing 34: Ejemplo 3

```
                        MENU
1.- Mostrar los n meros pares existentes desde 200 hasta 300.
2.- Mostrar los n meros multiplos de 3 existentes desde 100 hasta
   200.
3.- Mostrar los n meros multiplos de 7 existentes en los 50 numeros.
4.- Salir del programa

Ingrese opcion: 3
119, 175, 357, 196, 399, 413, 259,
```

Listing 35: Ejemplo 4

```
                        MENU
1.- Mostrar los n meros pares existentes desde 200 hasta 300.
2.- Mostrar los n meros multiplos de 3 existentes desde 100 hasta
   200.
3.- Mostrar los n meros multiplos de 7 existentes en los 50 numeros.
4.- Salir del programa

Ingrese opcion: 4
$
```

12. Una empresa le ha contratado para dar mantenimiento a un programa creado por un antiguo progra- mador en C++. Usted usará sus recientes conocimientos en asignación dinamica de memoria para modificarlo y mejorar el uso de recursos del programa:

Listing 36: Ejemplo 2

```
#include <iostream>
#include <iomanip>
using namespace std;

void printArray(float *p, int n);

int main() {
  float *p = nullptr;

  int n = 0;
  cout << "El.numero.de.notas.a.registrar\n";
  cin >> n;
  float array1[n];
  for (int i=0; i<n; i++){
    cout << "ingrese.la.nota:";
    cin >> array1[i];
  }
  p = array1;
  // imprimir el promedio
  printArray(p, n);
  return 0;
}
```

```
void printArray(float *p, int n){
  float suma = 0.0, promedio = 0.0;

  for (int i=0; i<n; i++)
    suma += p[i];
  promedio = suma*1.0/(n);
  cout << "El.promedio.es:" << fixed << setprecision(4)<< promedio;

}
```

- Modifique donde corresponda para crear un arreglo con memoria d´ınamica.
- Evalue si debe modificar la funcion que imprime el promedio
- Reemplace variables estaticas por dinámicas en el programa main.

13. Un arreglo o array es una serie de elementos del mismo tipo colocados en ubicaciones de memoria contiguas a las que se puede hacer referencia individualmente agregando un ´ındice a un identificador único. Escribe un programa que permita al usuario ingresar la cantidad de elementos que tendrá el arreglo, luego asigna un número aleatorio entre 0 y 20 a cada elemento del arreglo. Finalmente muestra los tres valores valores ingresados más altos.

    Algunos ejemplos de diálogo de este programa serían:

Listing 37: Ejemplo 1

```
Ingrese la cantidad de elementos:7
Los numeros generados entre 0 y 20 son:
7, 7, 2, 20, 13, 20, 9
Los 3 numeros mas altos son:
20
20
13
```

Listing 38: Ejemplo 2

```
Ingrese la cantidad de elementos:10
Los numeros generados entre 0 y 20 son:
5, 18, 7, 9, 13, 6, 19, 19, 17, 7
Los 3 numeros mas altos son:
19
19
18
```

14. Crear un programa que utilice ASIGNACION DINÁMICA DE MEMORIA y que permita crear un array de N elementos:

- El valor de N debe ser ingresado desde teclado
- Rellenar el array con números aleatorios, el rango a generar es de 1 a 100.
- El programa debe copiar los números pares al array1 y los impares al array2
- Mostrar en pantalla las variables ARRAY, ARRAY1 Y ARRAY2

Algunos ejemplos de diálogo de este programa serían:

Listing 39: Ejemplo 1

```
N: 10
array: 54,36,2,88,5,69,98,55,14,17
array1: 54,36,2,88,98,14
array2: 5,69,55,17
```

15. Elabore un programa que solicite al usuario el ingreso de un número F y un número C. Donde F representa el número de filas y C el número de columnas de la matriz dinámica a crear. Validar que F sea mayor a 5 y C sea un número entero par. Se llenará con valores numéricos aleatorios, luego de lo cual se generan dos nuevas matrices que resultan de dividir verticalmente la matriz inicial. Imprimar las dos nuevas matrices.

Algunos ejemplos de diálogo de este programa serían:

Listing 40: Ejemplo 1

```
Ingrese cantidad de filas:4
Ingrese cantidad de filas:5
Ingrese cantidad de filas:6
Ingrese cantidad de columnas:6

Matriz original:
       81          54          68          54          32          61
       32          71          51          67          44          60
       43          34          56          84          55          90
       41          24          91          27           9          92
       25          50          69          26          90          43
       81           6           2          21           3          54

Matriz 1:
       81          54          68
       32          71          51
       43          34          56
       41          24          91
       25          50          69
       81           6           2

Matriz 2:
       54          32          61
       67          44          60
       84          55          90
       27           9          92
       26          90          43
       21           3          54

Process finished with exit code 0
```

Listing 41: Ejemplo 2

```
Ingrese cantidad de filas:7
```

```
7
Ingrese cantidad de columnas:33
Ingrese cantidad de columnas:77
Ingrese cantidad de columnas:88
Matriz original:
51 59 13 95 47 42 14 26
51 98 94 56 37 96 51 64
28 19 97 51 33 49 82 3
37 81 82 10 39 56 75 36
7 54 80 17 72 19 25 9
73 22 47 74 72 96 6 47
31 12 18 45 81 49 61 6

Matriz 1:
51 59 13 95
51 98 94 56
28 19 97 51
37 81 82 10
7 54 80 17
73 22 47 74
31 12 18 45

Matriz 2:
47 42 14 26
37 96 51 64
33 49 82 3
39 56 75 36
72 19 25 9
72 96 6 47
81 49 61 6

Process finished with exit code 0
```

Listing 42: Ejemplo 3

```
Ingrese cantidad de filas:8
Ingrese cantidad de columnas:9
Ingrese cantidad de columnas:4

Matriz original:
94 73 75 63
28 16 79 62
35 0 75 57
81 41 79 29
61 34 3 68
43 92 47 48
16 72 42 18
```

```
        9          36          26          40

Matriz 1:
        94          73
        28          16
        35           0
        81          41
        61          34
        43          92
        16          72
         9          36

Matriz  2:
        75          63
        79          62
        75          57
        79          29
         3          68
        47          48
        42          18
        26          40


Process finished with exit code 0
```

16. Alessia es una niña muy afortunada y se ganó un nuevo juguete **"Don't Break the Ice"**, que es un juego de mesa y se juega con un conjunto de "bloques de hielo" de plástico (generalmente 6x6), un mazo en miniatura y un soporte. Un bloque de hielo es más grande que el resto (Normalmente 2x2), y ya sea un hombre, un oso polar o un Pingüino, se encuentra en este bloque. Los jugadores se turnan para quitar bloques tocando con el mazo. El juego termina cuando un jugador que "rompe el hielo", lo que provoca que el hombre, el oso o Pingüino se caiga. El jugador que haya eliminado la mayoría de los bloques sin "romper el hielo" es el ganador.

Figure 2: Juego "Don't Break the Ice"

Alessia, decide experimentar con su nuevo juguete y extrañamente cuando golpea el bloque grande los demás bloques se caen, pero, cuando golpea un bloque pequeño se cae solo el bloque golpeado.
Desarrolle un algoritmo que permita modelar las ocurrencias de Alessia. Para ello, considere uti- lizar una matriz dinámica de 6x6, además, cada bloque pequeño estará representado por cualquier número mayor a 2 o menor a 10, el bloque grande estará representado por 1s, los bloques caídos serán representados con 0.

Input

- La primera línea contiene los elementos de la matriz 6x6 que son números enteros.

3	3	6	7	8	8
5	7	8	**1**	**1**	3
6	9	5	**1**	**1**	8
7	6	9	3	8	8
3	3	6	7	7	8
3	3	6	7	8	3

- La segunda línea contiene dos enteros positivos F y C ($0 \leq F \leq 6; 0 \leq C \leq 6$), lo que correspondea la ubicación del bloque a golpear.

Ouput

- Se espera Imprimir una nueva matriz con los bloques caídos.Algunos

ejemplos de este programa serían:

Listing 43: Ejemplo 1

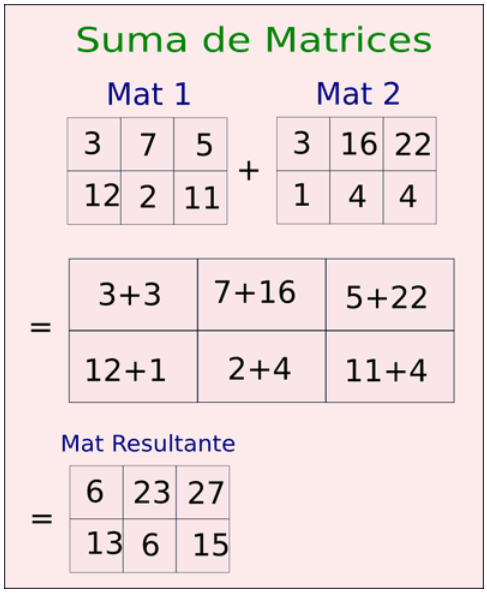
```
Ingrese ubicacion del bloque: 0 2
Output:

3 3 0 7 8 8
5 7 8 1 1 3
6 9 5 1 1 8
7 6 9 3 8 8
3 3 6 7 7 8
3 3 6 7 8 3
```

Listing 44: Ejemplo 2

```
Ingrese ubicacion del bloque: 1 4
Output:

0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Listing 45: Ejemplo 3

```
Ingrese ubicacion del bloque: 2 3
Output:

0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```



17. Elabore un algoritmo que permita sumar dos matrices y almacenar el resultado en una nueva matriz,tal como se muestra en la siguiente imagen:

Las funciones a implementar son las que se indica en la función main:

```
#include <iostream>
using namespace std;
int main()
{
        int n, m;
        cin>>n>>m;//dimension de la matriz
        int **ma1 = llenar(n, m);//llenar desde teclado
        int **ma2 = llenar(n, m);//llenar desde teclado
        int **maTsum = suma(ma1, ma2, n, m);
        cout<<"---\n";
        mostrar(maTsum, n, m);
        return 0;
}
```

18. Dado un programa que genera una matriz cuadrada de forma dinamica de valores entre 0 y 9 comose muestra en el código, corregir y completar el codigo faltante en la función **main** y desarrollar una función **es triangulo superior** (declaración y definición) que determine si la matriz es triangular superior. Se dice que una matriz es triangular superior si todos los valores por debajo de la diagonal principal son cero, Ejemplo:

```
1   4   5   6
0   2   7   8
0   0   9   3
0   0   0   2

2   2   2   1
0   2   2   2
0   0   2   2
0   0   0   2
```

Listing 46: Codigo

```
#include <iostream>
#include "matrices.h"

int main() {
    range n = 0;
    cout >> "tamano:."; cin >> n;

    number ** matriz = generar_matriz(nfil, ncol);
    actualizar_matriz(matriz, nfil, ncol);
    imprimir_matriz(matriz, nfil, ncol);

    if (es_triangulo_superior(matriz, nfil));
}
```

Algunos ejemplos de diálogo de este programa serían:

Listing 47: Ejemplo 1

```
tamano: 3
1   3   4
0   4   0
0   0   7
Es  tri angular Superior.
```

Listing 48: Ejemplo 1

```
tamano:   4
1   3   4   6
0   4   5   4
0   0   7   6
0   4   0   2
No  es  tri angular Superior.
```

19. Dado una matriz bidimensional de tamaño 5 filas y 7 columnas generada de forma dinámica. Realizar las siguientes operaciones:

    1. Llenar la matriz con números aleatorios entre -100 y 100.
    2. Mostrar la matriz generada.
    3. Mostrar la matriz con todos los números negativos existente. Mostrar ”-” en caso no cumpla la condición.
    4. Mostrar la matriz con todos los números positivos existentes. Mostrar ”-” en caso no cumpla la condición.

    Tomar en cuenta lo siguiente:

    - Realizar la asignación en la memoria de forma dinámica.
    - Compilación: *g++ -c main.cpp funciones.h implementacion.cpp*

    Algunos ejemplos de diálogo de este programa serían:

Listing 49: Ejemplo 1

```
Matriz generada:
        -9        -48        -96         63         97         17         35
        21        -99         26        -73         21        -24         16
       -67         24        -71         42         91        -28         81
        36        -64         41         -6        -50          9        -92
        44        -91         99        -65        -87        -97         50
Mostrar matriz con números negativos:
        -9        -48        -96          -          -          -          -
         -        -99          -        -73          -        -24          -
       -67          -        -71          -          -        -28          -
         -        -64          -         -6        -50          -        -92
         -        -91          -        -65        -87        -97          -
Mostrar matriz con números positivos:
         -          -          -         63         97         17         35
        21          -         26          -         21          -         16
         -         24          -         42         91          -         81
        36          -         41          -          -          9          -
        44          -         99          -          -          -         50
```

Listing 50: Ejemplo 2

```
Matriz generada:
         7         29         61        -34         72        -21        -73
         1         21         50        -65         59        -79        -36
       -31        -80         39        -75         67         76         85
       -51         94         -8        -12         94          2         -4
       -90         63        -61        -31        -55       -100         36
Mostrar matriz con números negativos:
         -          -          -        -34          -        -21        -73
         -          -          -        -65          -        -79        -36
       -31        -80          -        -75          -          -          -
       -51          -         -8        -12          -          -         -4
       -90          -        -61        -31        -55       -100          -
Mostrar matriz con números positivos:
         7         29         61          -         72          -          -
         1         21         50          -         59          -          -
         -          -         39          -         67         76         85
         -         94          -          -         94          2          -
         -         63          -          -          -          -         36
```

Listing 51: Ejemplo 3

```
Matriz generada:
        33         11         87         44        -39         26        -90
        26        -27        -86         20        -77         64         65
        92         66        -14         83         50         30         74
       -90        -34         85        -50         70         -2        -47
        61        -76         30        -54         36        -31         91
Mostrar matriz con números negativos:
         -          -          -          -        -39          -        -90
```

```
        -       -27       -86         -       -77         -         -
        -         -       -14         -         -         -         -
      -90       -34         -       -50         -        -2       -47
        -       -76         -       -54         -       -31         -
Mostrar matriz  con  n  meros  positivos:
       33        11        87        44         -        26         -
       26         -         -        20         -        64        65
       92        66         -        83        50        30        74
        -         -        85         -        70         -         -
       61         -        30         -        36         -        91
```

Listing 52: Ejemplo 4

```
 Matriz generada:
       71        17        73        81       -45        57       -80
       50       -37        77        41        16       -78       -76
      -15        45       -47        38       -35        31        47
      -27       -99         1        -6       -44       -54        33
       89        87        -2        13         4       -77        -6
Mostrar  matriz con  n meros  negativos:
        -         -         -         -       -45         -       -80
        -       -37         -         -         -       -78       -76
      -15         -       -47         -       -35         -         -
      -27       -99         -        -6       -44       -54         -
        -         -        -2         -         -       -77        -6
Mostrar  matriz con  n meros  positivos:
       71        17        73        81         -        57         -
       50         -        77        41        16         -         -
        -        45         -        38         -        31        47
        -         -         1         -         -         -        33
       89        87         -        13         4         -         -
```

20. Usted tiene información de 3 meses de la cantidad de ventas de 3 productos de la canasta basica de alimentos:

producto	enero	febrero	marzo
leche	10	20	30
huevos	30	20	10
pan	5	10	15

- Usando arreglos con memoria dinamica haga un programa para leer los datos.
- Lea precios por cada mes.
- Imprima los promedios por cada producto.
- Libere los recursos usados por el array.

Algunos ejemplos de diálogo de este programa serían:

Listing 53: Ejemplo 1

```
Ingrese precio (leche): 10
Ingrese precio (leche): 20
```

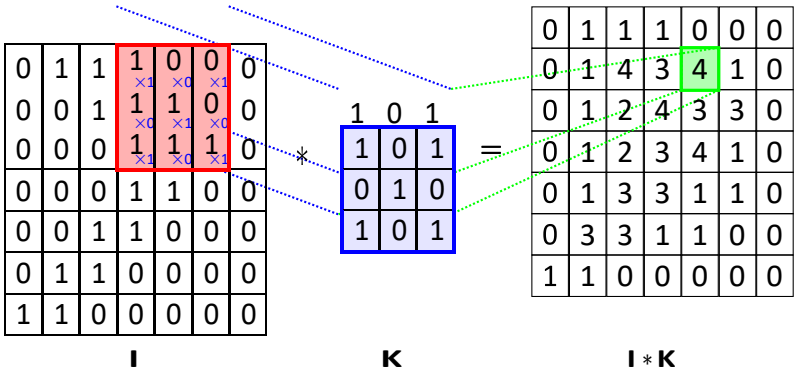
```
Ingrese precio (leche): 30
Ingrese precio (huevo): 30
Ingrese precio (huevo): 20
Ingrese precio (huevo): 10
Ingrese precio (pan): 5
Ingrese precio (pan): 10
Ingrese precio (pan): 15

Promedio leche:20
Promedio huevo:20
Promedio pan: 10
```

21. La mayoria de los filtros que usas en Facebook o Instagram usan matrices de convolución. Convolución es el tratamiento de una matriz por otra que se llama "kernel" K. El filtro matriz de convolución usa una primera matriz que es la imagen que será tratada **I**. La imagen es una colección bidimensional de píxeles en coordenada rectágular. El kernel usado depende del efecto deseado y los de tamaño de 3*x*3, son las más usadas y son suficiente para los efectos deseados.

    Escribe un programa que permita al usuario ingresar las dimensiones de la matriz **I**. Luego el programadeberá crear de manera dinámica la matriz con elementos aleatorios entre cero y uno. Después de esto deberás aplicar el filtro de convolucion K tal como se muestra en la imagen y generar una nueva matriz **IK**. Los elementos de la nueva matriz **IK** serán calculados usando el filtro de convolución. Por ejemplo el elemento **IK**[**1**][**4**] será calculado de la siguiente forma:

$$\begin{aligned}\mathbf{IK}[1][4] = {} & \mathbf{I}[1][3]\cdot\mathbf{K}[0][0] + \mathbf{I}[1][4]\cdot\mathbf{K}[0][1] + \mathbf{I}[1][5]\cdot\mathbf{K}[\mathbf{0}][\mathbf{2}] + \\ & \mathbf{I}[2][3]\cdot\mathbf{K}[1][0] + \mathbf{I}[2][4]\cdot\mathbf{K}[1][1] + \mathbf{I}[2][5]\cdot\mathbf{K}[\mathbf{1}][\mathbf{2}] + \\ & \mathbf{I}[3][3]\cdot\mathbf{K}[2][0] + \mathbf{I}[3][4]\cdot\mathbf{K}[2][1] + \mathbf{I}[3][5]\cdot\mathbf{K}[\mathbf{2}][\mathbf{2}]+\end{aligned} \tag{1}$$

    Hint: Ten en cuenta que los elementos del borde de la matriz no son tomados en cuenta.



    Algunos ejemplos de diálogo de este programa serían:

    Listing 54: Ejemplo 1

```
Ingrese la cantidad de filas:5
Ingrese la cantidad de columnas:7
I:
1 0 1 1 0 1 0
1 1 1 1 1 0 1
0 1 0 1 1 1 0
1 1 0 1 1 1 1
1 1 1 1 0 1 1
```

```
K:
1          0          1
0          1          0
1          0          1
IK:
1          0          1          1          0          1          0
1          3          4          3          5          1          1
0          4          4          4          4          5          0
1          3          4          3          5          3          1
1          1          1          1          0          1          1
```

Listing 55: Ejemplo 2

```
Ingrese la cantidad de filas:4
Ingrese la cantidad de columnas :3
I:
1          0          0
1          0          1
1          1          1
1          1          0
K:
1          0          1
0          1          0
1          0          1
IK:
1          0          0
1          3          1
1          4          1
1          1          0
```

Listing 56: Ejemplo 3

```
Ingrese la cantidad de filas:2
Ingrese la cantidad de columnas :2
I:
1          0
0          0
K:
1          0          1
0          1          0
1          0          1
IK:
1          0
0          0
```

22. Crear un programa que utilice ASIGNACION DINÁMICA DE MEMORIA y que permita crear una matriz de N x M:

- Los valores de N (filas) y M (columnas) deben ser ingresados desde teclado
- Rellenar la matriz con números aleatorios, el rango a generar es de 1 a 10

- Mostrar la matriz en pantalla
- Indicar es la suma de las COLUMNAS que tiene el m´ınimo valor, si hay varias columnas quetienen el m´ınimo valor, debe indicarlas

Algunos ejemplos de diálogo de este programa serían:

Listing 57: Ejemplo 1

```
N=3
M=4
1       3       3       7
5       4       1       0
1       0       9       2

Respuesta
Minimo Valor: 7
COLUMNAS: 0, 1
```