



ACTIVIDAD 01: TIPOS DE DATO PRIMITIVOS Y TIPOS DE DATO ESTRUCTURADOS

ESTUDIANTE: EFRAIN ROBLES PULIDO

CODIGO: 221350095

NOMBRE DE LA MATERIA: ESTRUCTURAS DE DATOS I

SECCIÓN: D12

CLAVE: I5886

FECHA: DOMINGO 23 DE ENERO DE 2022

Tabla de autoevaluación:

Autoevaluación			
Concepto	Sí	No	Acumulación
Bajé el trabajo de internet o alguien me lo pasó (aunque sea de forma parcial)	-100 pts	0 pts	-5 pts
Incluí el código fuente en formato de texto (sólo si funciona cumpliendo todos los requerimientos)	+25 pts	0 pts	25 pts
Incluí las impresiones de pantalla (sólo si funciona cumpliendo todos los requerimientos)	+25 pts	0 pts	25 pts
Incluí una portada que identifica mi trabajo (nombre, código, materia, fecha, título)	+25 pts	0 pts	25 pts
Incluí una descripción y conclusiones de mi trabajo	+25 pts	0 pts	25 pts
Suma:			95 pts

Introducción:

Para comenzar se tuvo que leer apropiadamente las instrucciones para crear primero una función para el menú donde nos indicara que letra se debe introducir para mostrar los tipos de datos primitivos o las matrices. Se utilizo el ciclo do-while para que si es diferente a "c" se repita el programa.

Si es "a" se deberá mostrar como en forma de tabla, el tamaño de bits y los valores máximos y mínimos de las variables primitivas mediante las funciones del programa como sizeof y los numeric_limits.

Si es "b" se deberá de pedir el tamaño de las matrices con rango de 3 a 10 que estará en un ciclo do-while para confirmar que este en el rango acordado, si cumple, se crearán matrices A,B,C,D como objetos, la matriz A y la matriz B tendran valores reales aleatorios de -100 a 100 con la librería random, para después hacer que la matriz C sea el calculo la multiplicación de las matrices A y B mediante una triple anidación del ciclo for y la sobrecarga de los operadores del *, después hacer que la matriz D que será la suma de las matrices A y B que es directamente la suma de las matrices normales dentro de dos for anidados y la sobrecarga de operadores del +, también se utilizara hacer una sobrecarga del operador = para el correcto funcionamiento del código, finalmente se mostrara las matrices apropiadamente con 2 decimales, utilizando también 2 for anidados para mostrarlos.

Código fuente:

```
#include <iostream>
#include <string>
#include <random>
#include <chrono>
#include <functional>

using namespace std;
class Matriz {
    private:
        int sizeM; //Atributos
        float matriz[10][10];

    public:
        Matriz() {} //Constructores
        Matriz(const Matriz& mtrz) : sizeM(mtrz.sizeM) {
            for(int i = 0; i < this->sizeM; i++) {
                for(int j = 0; j < this->sizeM; j++) {
                    this->matriz[i][j] = mtrz.matriz[i][j];
                }
            }
        }

        Matriz(const int& N) : sizeM(N) {}

        //Metodos
        void fillMatriz() { //Llena matriz
            std::default_random_engine generator(clock()); //Llama
            //generador de numero reales random
            std::uniform_real_distribution<double> distribution(-
100.00,100.00);
            for(int i = 0; i < sizeM; i++) {
                for(int j = 0; j < sizeM; j++) {
                    matriz[i][j] = distribution(generator);
                }
            }
        }

        string printMatriz() const { //Imprime matriz resultante
            string _space;
            char myStr[11];
            for(int i = 0; i < this->sizeM; i++) {
                for(int j = 0; j < this->sizeM; j++) {
                    sprintf(myStr,"%8.2f", this->matriz[i][j]);
                    _space+= myStr;
                    _space+= "\t";
                }
                _space+= "\n";
            }
            return _space;
        }

        //Operadores
        Matriz operator + (const Matriz& mtrz) {
            Matriz resultante(sizeM);
```

```

        for(int i = 0; i < this->sizeM; i++) {
            for(int j = 0; j < this->sizeM; j++) {
                resultante.matriz[i][j] = matriz[i][j] +
mtrz.matriz[i][j];
            }
        }
        return resultante;
    }

Matriz operator * (const Matriz& mtrz) {
    Matriz resultante(sizeM);
    for(int i = 0; i < sizeM; i++) {
        for(int j = 0; j < sizeM; j++) {
            float sum = 0;
            for(int a = 0; a < sizeM; a++) {
                sum += matriz[i][a] * mtrz.matriz[a][j];
            }
            resultante.matriz[i][j] = sum;
        }
    }
    return resultante;
}

Matriz& operator = (const Matriz& mtrz) {
    for(int i = 0; i < sizeM; i++) {
        for(int j = 0; j < sizeM; j++) {
            matriz[i][j] = mtrz.matriz[i][j];
        }
    }
    return *this;
}

};

class TiposDatosPrimitivos {
public:
    void InfoData() {
        cout<<"\n  Tipo de dato          Bits          Valor minimo
Valor maximo"<<endl;
        cout<<"-----"
-----"<<endl;
        std::cout<<"Caracter Con Signo          "<<sizeof(signed
char)*8<<"          "<<+numeric_limits<signed char>::min()<<"
"<<+numeric_limits<signed char>::max()<<endl;
        cout<<"-----"
-----"<<endl;
        std::cout<<"Caracter Sin Signo          "<<sizeof(unsigned
char)*8<<"          "<<+numeric_limits<unsigned char>::min()<<"
"<<+numeric_limits<unsigned char>::max()<<endl;
        cout<<"-----"
-----"<<endl;
        std::cout<<"Entero Corto Con Signo          "<<sizeof(signed
short int)*8<<"          "<<+numeric_limits<signed short int>::min()<<"
"<<+numeric_limits<signed short int>::max()<<endl;
        cout<<"-----"
-----"<<endl;
        std::cout<<"Entero Corto Sin Signo          "<<sizeof(unsigned
short int)*8<<"          "<<+numeric_limits<unsigned short

```

```

int>::min()<<"                                "<<+numeric_limits<unsigned short
int>::max()<<endl;
    cout<<"-----"
-----"<<endl;
    std::cout<<"Entero Largo Con Signo          "<<sizeof(signed
long int)*8<<"                                "<<+numeric_limits<signed long int>::min()<<"
"<<+numeric_limits<signed long int>::max()<<endl;
    cout<<"-----"
-----"<<endl;
    std::cout<<"Entero Largo Sin Signo          "<<sizeof(unsigned
long int)*8<<"                                "<<+numeric_limits<unsigned long
int>::min()<<"                                "<<+numeric_limits<unsigned long
int>::max()<<endl;
    cout<<"-----"
-----"<<endl;
    std::cout<<"Real De Precision Simple
"<<sizeof(float)*8<<"                                "<<+numeric_limits<float>::min()<<"
"<<+numeric_limits<float>::max()<<endl;
    cout<<"-----"
-----"<<endl;
    std::cout<<"Real De Doble Precision
"<<sizeof(double)*8<<"                                "<<+numeric_limits<double>::min()<<"
"<<+numeric_limits<double>::max()<<endl;
    cout<<"-----"
-----"\\n"<<endl;

    }

};

void menu() {
    cout<<"\\tSeleccione una opcion:"<<endl;
    cout<<"a) Tamaño y rangos de los Tipos de Dato Primitivos"<<endl;
    cout<<"b) Ejemplo de uso de Tipo de dato Estructurado"<<endl;
    cout<<"c) Salir\\n"<<endl;
}

int main() {
    char opcMenu;
    do {
        menu();
        cin>>opcMenu;

        if(opcMenu=='a') {
            TiposDatosPrimitivos ejemplo;
            ejemplo.InfoData();

        }
        else if(opcMenu=='b') {
            int N = 0;
            do {
                cout<<"Tamaño de matriz (3 a 10): "<<endl;
                cin>>N;
            }
            while((N<3) || (N>10));

            Matriz A(N), B(N), C(N), D(N);
            A.fillMatriz();
            cout << "Matriz A:" << endl;

```

```

        cout << A.printMatriz() << endl << endl;

        B.fillMatriz();
        cout << "Matriz B:" << endl;
        cout << B.printMatriz() << endl << endl;

        C = A * B;
        cout << "Matriz C - Multiplicacion de las matrices:" << endl;
        cout << C.printMatriz() << endl << endl;

        D = A + B;
        cout << "Matriz D - Suma de las matrices:" << endl;
        cout << D.printMatriz() << endl << endl;

    }
}

while(opcMenu!='c');

cout<<"Fin del programa\n"<<endl;
return 0;
}

```

"E:\Documentos PC\UDG Materias\estructura\Tarea 1\Tarea 1.exe"

Seleccione una opción:
a) Tamaño y rangos de los Tipos de Dato Primitivos
b) Ejemplo de uso de Tipo de dato Estructurado
c) Salir

a

Tipo de dato	Bits	Valor minimo	Valor maximo
Caracter Con Signo	8	-128	127
Caracter Sin Signo	8	0	255
Entero Corto Con Signo	16	-32768	32767
Entero Corto Sin Signo	16	0	65535
Entero Largo Con Signo	32	-2147483648	2147483647
Entero Largo Sin Signo	32	0	4294967295
Real De Precision Simple	32	1.17549e-38	3.40282e+38
Real De Doble Precision	64	2.22507e-308	1.79769e+308

Seleccione una opción:
a) Tamaño y rangos de los Tipos de Dato Primitivos
b) Ejemplo de uso de Tipo de dato Estructurado
c) Salir

b

Tamaño de matriz (3 a 10):
4

Matriz A:

7.35	91.39	15.91	28.05
-4.63	-19.52	-82.08	-19.53
-14.28	97.39	-82.21	56.45
85.54	-29.20	-62.27	61.37

Matriz B:

70.42	8.69	53.83	-14.22
64.76	19.31	-12.94	39.87
1.11	31.08	91.33	-82.68
-60.60	78.64	-59.88	-14.23

Matriz C - Multiplicacion de las matrices:

4753.22	4529.12	-1013.23	1824.23
-497.67	-4503.86	-6323.38	6351.84
1788.86	3640.75	-12917.79	10080.17
344.55	3070.54	-4379.54	1894.15

Matriz D - Suma de las matrices:

77.77	100.08	69.74	13.83
60.13	-0.21	-95.02	20.34
-13.17	128.47	9.12	-26.23
24.94	49.44	-122.15	47.14

```
"E:\Documentos PC\UDG Materias\estructura\Tarea 1\Tarea 1.exe"
4753.22      4529.12      -1013.23      1824.23
-497.67      -4503.86      -6323.38      6351.84
1788.86      3640.75      -12917.79     10080.17
344.55       3070.54      -4379.54      1894.15

Matriz D - Suma de las matrices:
  77.77      100.08      69.74      13.83
  60.13      -0.21      -95.02      20.34
 -13.17      128.47      9.12      -26.23
  24.94      49.44      -122.15     47.14

      Seleccione una opcion:
a) Tamano y rangos de los Tipos de Dato Primitivos
b) Ejemplo de uso de Tipo de dato Estructurado
c) Salir

c
Fin del programa

Process returned 0 (0x0)   execution time : 1489.441 s
Press any key to continue.
```

Conclusión:

En esta práctica fue difícil de hacer debido que tuve que investigar cómo se debe de programar en POO ya que en las siguientes practicas tendremos que usarlo muy seguido, primero hice el programa como me enseñaron que es de la forma estructurada, una vez me salió el programa, lo pase a POO en donde tuve que usar las clases para poner sus atributos(las variables a utilizar), construcciones(como se inicializarían los atributos) y métodos(las acciones que harán nuestros objetos) para poder crear los objetos que serían mis matrices, también me fue difícil entender como era la sobrecarga de operadores debido a que se sobrescribe los operadores del mismo programa para poder ajustarlas a nuestras necesidades pero con mucha investigación y pruebas pude realizar el código. Además, también tuve que investigar cómo se sacar los números aleatorios para las matrices que se utilizó funciones de la librería random y chrono para que si fueran aleatorios, también se utilizó la librería limits para poder usar sus funciones para encontrar el valor máximo y mínimo de las variables primitivas y el sizeof() * 8 para encontrar la cantidad de bits que usan. Tengo mucho de que aprender sobre POO debido que es una forma muy diferente a como me enseñaron programar.