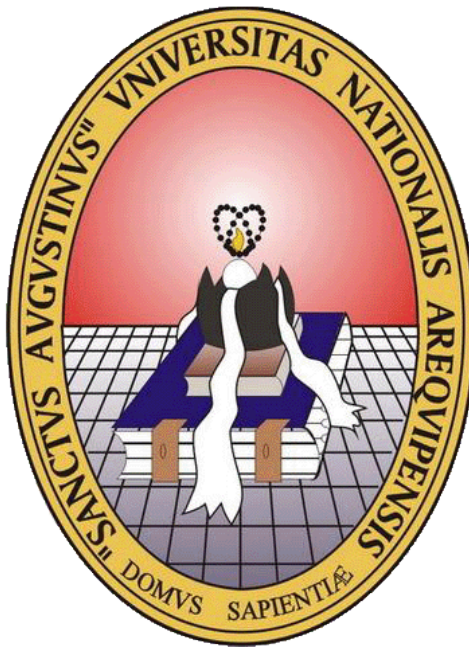


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN



PRÁCTICA DE LABORATORIO 19
CURSO DE CIENCIAS DE LA COMPUTACIÓN II

ESTUDIANTE:
RUIZ MAMANI, EDUARDO GERMÁN

EMAIL: eruizm@unsa.edu.pe

CUI: 20193061

TURNO:

C

AREQUIPA- PERÚ

2021

LINK DEL REPOSITORIO: https://github.com/EGRM23/CCII_20193061.git

1. EJERCICIO

1. Suma los dígitos de un numero de forma recursiva utilizando metaprogramación.

- **CÓDIGO**

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

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

class RegrLineal {
public:
    vector<double> operator()(vector<int> x, vector<int> y) {
        double sum1 = 0, sum2 = 0, medx = 0, medy = 0;
        double a, b;

        for(int i = 0; i < (int)x.size(); i++) {
            medx += x[i];
            medy += y[i];
        }

        medx = medx/x.size();
        medy = medy/y.size();

        for(int i = 0; i < (int)x.size(); i++) {
            sum1 += (x[i]-medx)*(y[i]-medy);
            sum2 += (x[i]-medx)*(x[i]-medx);
        }

        b = sum1/sum2;
        a = medy - b*medx;

        vector<double> result = {a,b};

        return result;
    }

    void imprimir(vector<int> x) {
        cout << "[ ";
        for(int i = 0; i < (int)x.size(); i++) {
            cout << x[i] << " ";
        }
        cout << "]\n";
    }
};

int main(int argc, char *argv[]) {
    vector<int> x = {-13,0,9,20,33,50};
    vector<int> y = {322,335,337,346,352,365};

    RegrLineal Ej1;
    vector<double> param = Ej1(x,y);
```

```

        cout << "x = ";
        Ej1.imprimir(x);
        cout << "y = ";
        Ej1.imprimir(y);
        cout << endl;
        cout << "y = " << param[0] << " + " << param[1] << "x";
        return 0;
    }

```

- **CAPTURAS**

```

C:\Program Files (x86)\Zinjal\bin\runner.exe
x = [ -13 0 9 20 33 50 ]
y = [ 322 335 337 346 352 365 ]

y = 332.115 + 0.649587x

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>_

```

2. EJERCICIO

2. Calcular el valor de la posición fibonacci usando recursividad utilizando metaprogramación.

- **CÓDIGO**

```

#include <iostream>
#include <vector>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

class Buscar {
    vector<int> ref =
{1,3,6,9,2,2,2,5,4,5,6,8,9,2,1,2,3,4,6,3,3,3,3,5,6,7};
public:
    vector<int> operator()(int ini, int fin, int v) {
        int cont = 0;
        vector<int> result = {};
        for (int i = ini; i <= fin; i++) {
            if (ref[i] == v) {
                cont++;
                result.push_back(i);
            }
        }
        return result;
    }
    void imprimir(vector<int> x) {
        cout << "[ ";
        for(int i = 0; i < (int)x.size(); i++) {
            cout << x[i] << " ";
        }
        cout << "]\n";
    }
}

```

```

        void mostrar() {
            cout << "v = ";
            imprimir(this->ref);
        }
    };

    int main(int argc, char *argv[]) {
        int posini = 0, posfin = 5, tb = 2;
        Buscar find1;
        vector<int> pos = find1(posini, posfin, tb);

        find1.mostrar();
        cout << "Posicion inicial: " << posini << endl;
        cout << "Posicion final: " << posfin << endl;
        cout << "Termino buscado: " << tb << endl;
        cout << "Coincidencias = ";
        find1.imprimir(pos);
        return 0;
    }

```

- **CAPTURAS**

```

C:\Program Files (x86)\Zinjal\bin\runner.exe
v = [ 1 3 6 9 2 2 2 5 4 5 6 8 9 2 1 2 3 4 6 3 3 3 3 5 6 7 ]
Posicion inicial: 0
Posicion final: 5
Termino buscado: 2
Coincidencias = [ 4 5 ]

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>_

```

3. EJERCICIO

3. Calcula la potencia de un numero de forma recursiva utilizando metaprogramación.

- **CÓDIGO:**

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <stdlib.h>
#include <time.h>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

class par {
public:
    int a;
    int b;
    par(int x, int y) : a(x), b(y) {}
};

```

```

class Algoritmo {
public:
    bool operator()(const par& obj1, const par& obj2) {
        return obj1.a < obj2.b;
    }

    void imprimir(vector<par> ref) {
        cout << "[ ";
        for(int i = 0; i < (int)ref.size(); i++) {
            cout << "[" << ref[i].a << ", " << ref[i].b <<
"] ";
        }
        cout << "]\n";
    }
};

int main(int argc, char *argv[]) {
    srand(time(NULL));
    vector<par> ref = {};

    for (int i = 0; i < 20; i++)
        ref.push_back(par(1+rand()%(100-1),1+rand()%(100-
1)));

    Algoritmo functor;

    cout << "v = ";
    functor.imprimir(ref);

    cout << endl;

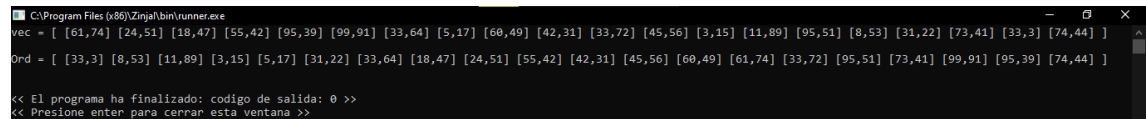
    cout << "Ord = ";
    sort(ref.begin(),ref.end(),functor);

    functor.imprimir(ref);

    return 0;
}

```

- **CAPTURAS**



```

C:\Program Files (x86)\Zingal\bin\runner.exe
vec = [ [61,74] [24,51] [18,47] [55,42] [95,39] [99,91] [33,64] [5,17] [60,49] [42,31] [33,72] [45,56] [3,15] [11,89] [95,51] [8,53] [31,22] [73,41] [33,3] [74,44] ]
Ord = [ [33,3] [8,53] [11,89] [3,15] [5,17] [31,22] [33,64] [18,47] [24,51] [55,42] [42,31] [45,56] [60,49] [61,74] [33,72] [95,51] [73,41] [99,91] [95,39] [74,44] ]
<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>

```