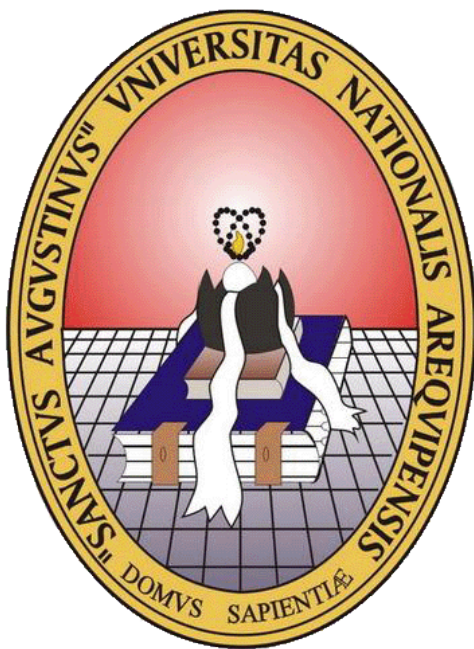


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN



PRÁCTICA DE LABORATORIO 18
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>
using namespace std;

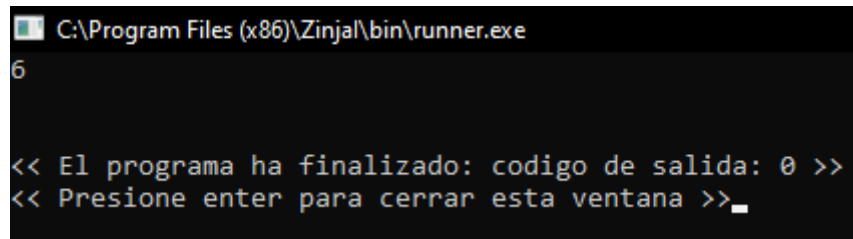
//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

template <int R>
struct sumadig {
    enum { suma = R - (R/10)*10 + sumadig<R/10>::suma };
};

template <>
struct sumadig<0> {
    enum { suma = 0 };
};

int main(int argc, char *argv[]) {
    int s = sumadig<123>::suma;
    cout << s << endl;
    return 0;
}
```

- **CAPTURAS**



2. EJERCICIO

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

- **CÓDIGO**

```
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

template <int R>
struct Fibonacci {
    enum { fibo = Fibonacci<R-1>::fibo + Fibonacci<R-2>::fibo };
};

template <>
```

```

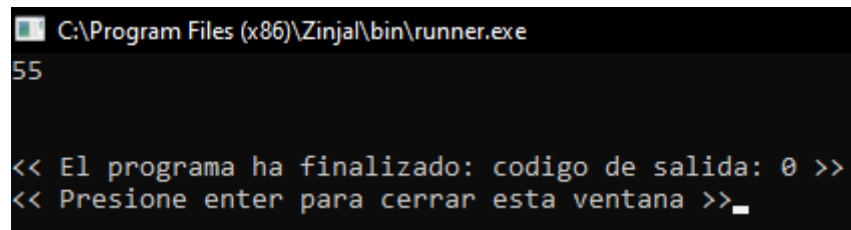
struct Fibonacci<1> {
    enum { fibo = 1 };
};

template <>
struct Fibonacci<2> {
    enum { fibo = 1 };
};

int main(int argc, char *argv[]) {
    int f = Fibonacci<10>::fibo;
    cout << f << endl;
    return 0;
}

```

- **CAPTURAS**



```

C:\Program Files (x86)\Zinjal\bin\runner.exe
55

<< 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>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

template <int R, int P>
struct Potencia {
    enum { result = R * Potencia<R,P-1>::result };
};

template <int R>
struct Potencia<R,0> {
    enum { result = 1 };
};

int main(int argc, char *argv[]) {
    int p = Potencia<2,10>::result;
    cout << p << endl;
    return 0;
}

```

- **CAPTURAS**

4. EJERCICIO

4. Construya una función recursiva que convierta un número decimal en una cadena que represente el valor del número en hexadecimal (base 16) utilizando metaprogramación.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- **CÓDIGO:**

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

//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061
const string molde = "0123456789ABCDEF";

template <int num, int base>
struct Convertidor {
    enum { nuevonum = num/base, dig = num%base };
    string result = Convertidor<nuevonum,base>().result +
molde[dig];
};

template <int base>
struct Convertidor<0,base> {
    string result = "";
};

int main(int argc, char *argv[]) {
    string cad = Convertidor<256+15,16>().result;
    cout << cad << endl;
    return 0;
}
```

- **CAPTURAS**

5. EJERCICIO

5. Ingresar un número y mostrar su equivalente en binario usando una función recursiva utilizando metaprogramación.

0 0 0 0	0 0	0	0 0
0 0 0 1	0 1	1	0 1
0 0 1 0	0 2	2	0 2
0 0 1 1	0 3	3	0 3
0 1 0 0	0 4	4	0 4
0 1 0 1	0 5	5	0 5
0 1 1 0	0 6	6	0 6
0 1 1 1	0 7	7	0 7
1 0 0 0	1 0	8	0 8
1 0 0 1	1 1	9	0 9
1 0 1 0	1 2	A	1 0
1 0 1 1	1 3	B	1 1
1 1 0 0	1 4	C	1 2
1 1 0 1	1 5	D	1 3
1 1 1 0	1 6	E	1 4
1 1 1 1	1 7	F	1 5

- **CÓDIGO:**

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

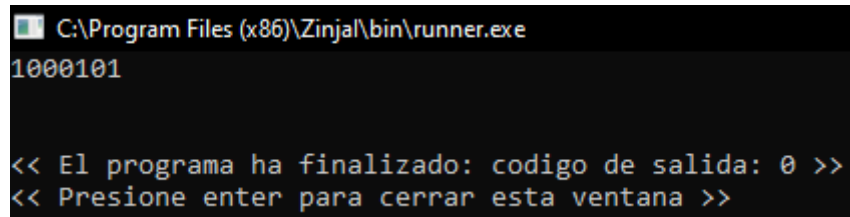
//EDUARDO GERMAN RUIZ MAMANI
//CUI 20193061

template <int num>
struct Convertidor {
    enum { result = num%2 + Convertidor<num/2>::result*10 };
};

template <>
struct Convertidor<0> {
    enum { result = 0 };
};

int main(int argc, char *argv[]) {
    int v = Convertidor<64+4+1>::result;
    cout << v << endl;
    return 0;
}
```

- **CAPTURAS**



```
C:\Program Files (x86)\Zinjal\bin\runner.exe
1000101

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