

Laboratorio 18

Metaprogramación

Nombre: Gabriel Fernando Rodriguez Cutimbo

CUI: 20212157

Grupo: B

Repositorio GitHub:

https://github.com/gaco123/EPCC_CCIL.git

1. Ejercicios

Resolver los siguientes ejercicios planteados:

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

Código:

```
/*  
1. Suma los dígitos de un numero de forma recursiva utilizando metaprogramación.  
*/  
  
#include <iostream>  
using namespace std;  
  
int sum_Dig(int x, int sum){  
    sum += x % 10;  
    if(x / 10 == 0){  
        return sum;  
    }  
  
    return sum_Dig(x / 10, sum);  
}  
  
int main(int argc, char *argv[]) {  
    int x;  
    cout << "PROGRAMA PARA CALCULAR LA SUMA DE DIGITOS DE UN NUMERO\n";  
    cout << "Ingrese un número: ";  
    cin >> x;  
    cout << "La suma de los digitos del número ingresado es: " << sum_Dig(x, 0);  
}
```

```
    return 0;  
}
```

Funcionamiento:

```
PROGRAMA PARA CALCULAR LA SUMA DE DIGITOS DE UN NUMERO  
Ingrese un número: 150  
La suma de los digitos del número ingresado es: 6
```

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

Código:

```
#include <iostream>  
#include <string>  
using namespace std;  
  
int pos_Fibonacci(int pos){  
    string error = "La secuencia no puede tener una posición menor o igual a 0\n\n";  
    try{  
        if(pos <= 0){  
            throw(error);  
        }  
    }  
    catch(string e){  
        cout << e;  
        exit(0);  
    }  
    if(pos == 1){  
        return 0;  
    }  
    else if(pos == 2){  
        return 1;  
    }  
    return pos_Fibonacci(pos - 2) + pos_Fibonacci(pos - 1);  
}  
  
int main(int argc, char *argv[]) {  
  
    int n1,n2;  
    cout << "PROGRAMA PARA CALCULAR LA SERIE DE FIBONACCI DESDE UNA POSICION  
INICIAL HASTA UNA N-POSICION\n\n";  
    cout << "Ingrese la posición inicial: ";  
    cin >> n1;  
    cout << "Ingrese la n-posición: ";  
    cin >> n2;
```

```
if(n1 != n2){
    for(int i = n1; i <= n2; i++){
        if(i == n1 && n1 != n2){
            cout << "[" << pos_Fibonacci(i) << ", ";
        }
        else if(i != n2){
            cout << pos_Fibonacci(i) << ", ";
        }
        else{
            cout << pos_Fibonacci(i) << "]";
        }
    }
}
else{
    cout << "[" << pos_Fibonacci(n1) << "]";
}

return 0;
}
```

Funcionamiento:

```
PROGRAMA PARA CALCULAR LA SERIE DE FIBONACCI DESDE UNA POSICION INICIAL HASTA UNA N-POSICION
Ingrese la posición inicial: 1
Ingrese la n-posición: 10
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

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

Código:

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

int my_pow(int base, int exponent){
    if(exponent == 1){
        return base;
    }
    else{
        base = base * my_pow(base,exponent-1);
        return base;
    }
}

int main(int argc, char *argv[]){
    int ba, exp;
    cout << "PROGRAMA PARA CALCULAR LA POTENCIA DE UN NUMERO\n";
    cout << "Ingrese la base: ";
```

```
cin >> ba;
cout << "Ingrese el exponente: ";
cin >> exp;
cout << "La potencia de " << ba << " con exponente " << exp << " es: " << my_pow(ba,exp);

return 0;
}
```

Funcionamiento:

```
PROGRAMA PARA CALCULAR LA POTENCIA DE UN NUMERO
Ingresa la base: 2
Ingresa el exponente: 5
La potencia de 2 con exponente 5 es: 32
```

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>
#include <string>
using namespace std;

string reverse_str(string& str){
    string temp;
    for(int i=str.length()-1; i>=0; i--){
        temp += str[i];
    }
    return temp;
}

string convert_Hex(int num, string& rpt){
    int x = num % 16;
    if(num == 0){
        if(rpt.length()==0){
            return "0";
        }
        return reverse_str(rpt);
    }
    else if(num == 10){
        rpt += "A";
    }
}
```

```
    }
    else if(num == 11){
        rpt += "B";
    }
    else if(num == 12){
        rpt += "C";
    }
    else if(num == 13){
        rpt += "D";
    }
    else if(num == 14){
        rpt += "E";
    }
    else if(num == 15){
        rpt += "F";
    }
    else{
        rpt += to_string(x);
    }

    return convert_Hex(num / 16, rpt);
}

int main(int argc, char *argv[]){

    int num;
    string rpt;
    cout << "PROGRAMA CONVERTIR UN NUMERO DECIMAL A UN NUMERO HEXADECIMAL\n";
    cout << "Ingrese un número: ";
    cin >> num;

    cout << "Número convertido a hexadecimal: ";
    cout << convert_Hex(num, rpt);

    return 0;
}
```

Funcionamiento:

```
PROGRAMA CONVERTIR UN NUMERO DECIMAL A UN NUMERO HEXADECIMAL
Ingrese un número: 160
Número convertido a hexadecimal: A0
```

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;

void convert_Bin(int n, string& rpt){
    if (n!=0){
        convert_Bin(n/2, rpt);
        rpt += to_string(n%2);
    }
}

int main(int argc, char *argv[]){

    int num;
    string rpt;
    cout << "PROGRAMA CONVERTIR UN NUMERO DECIMAL A UN NUMERO BINARIO\n";
    cout << "Ingrese un número: ";
    cin >> num;

    cout << "Número convertido a binario: ";
    convert_Bin(num, rpt);
    cout << rpt;

    return 0;
}
```

Funcionamiento:

```
PROGRAMA CONVERTIR UN NUMERO DECIMAL A UN NUMERO BINARIO
Ingrese un número: 100
Número convertido a binario: 1100100
```

2. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB18_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB18_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB18_GRUPO_A/B/C_CUI_ EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB18_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.