

Laboratorio 03

Funciones

Nombre: Gabriel Fernando Rodriguez Cutimbo

CUI: 20212157

Grupo: B

Repositorio GitHub:

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

1. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Hacer un programa que sin usar la función `pow()`, pero por medio de una función, calcule la potencia de un número (ambos números ingresados por teclado).

Código:

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

int pow_ex(int a, int b){
    int temp=a;
    for(int i=0; i<b-1; i++){
        a=a*temp;
    }
    return a;
}

int main(){
    int num1, num2;
    cout<<"PROGRAMA PARA CALCULAR LA POTENCIA DE UN NUMERO"<<endl;
    cout<<"Ingrese el número que será la base: ";
    cin>>num1;
    cout<<"Ingrese el número que será el exponente: ";
    cin>>num2;
    cout<<"El resultado de la potencia entre la base "<<num1<<" y el exponente "<<num2<<" es:
"<<pow_ex(num1,num2);

    return 0;
}
```

Funcionamiento:

```
PROGRAMA PARA CALCULAR LA POTENCIA DE UN NUMERO
Ingrese el número que será la base: 10
Ingrese el número que será el exponente: 3
El resultado de la potencia entre la base 10 y el exponente 3 es: 1000
```

2. Hacer un programa que lea por teclado un año, calcule y muestre si es bisiesto. Para realizar el cálculo utiliza una función llamada bisiesto. La función bisiesto recibe el año leído por teclado, comprueba si es o no bisiesto.

Código:

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

void leap_year(int a){
    if(a%4==0){
        cout<<"El año introducido SI es bisiesto";
    }
    else{
        cout<<"El año introducido NO es bisiesto";
    }
}

int main(){
    int year;
    cout<<"PROGRAMA PARA VERIFICAR SI UN AÑO ES BISIESTO"<<endl;
    cout<<"Ingrese el año a verificar: ";
    cin>>year;
    leap_year(year);

    return 0;
}
```

Funcionamiento:

```
PROGRAMA PARA VERIFICAR SI UN AÑO ES BISIESTO
Ingrese el año a verificar: 2004
El año introducido SI es bisiesto
```

3. Hacer un programa que lee por teclado la fecha actual y la fecha de nacimiento de una persona y por medio de una función calcule su edad en años, meses y días.

Código:

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

```
void actual_Age(string bird, string now){

    string temp;
    string actual;
    //Fecha de Nacimiento
    int yx;
    int mx;
    int dx;

    //Fecha Actual
    int yy;
    int my;
    int dy;

    //Edad Actual
    int yz=0;
    int mz=0;
    int dz=0;

    //Separación Nacimiento
    temp=bird.substr(0, 4);
    yx=stoi(temp);
    temp=bird.substr(5, 2);
    mx=stoi(temp);
    temp=bird.substr(8, 2);
    dx=stoi(temp);

    //Separación Fecha Actual
    temp=now.substr(0, 4);
    yy=stoi(temp);
    temp=now.substr(5, 2);
    my=stoi(temp);
    temp=now.substr(8, 2);
    dy=stoi(temp);

    //Calculo de Edad Actual en años, meses y días
    while(yy!=yx&&my!=mx&&dy!=dx){
        dx++;
        if(mx==1&&dx==32){
            mx++;
            dx=0;
        }
        if(mx==2&&dx==30){
            mx++;
            dx=0;
        }
        if(mx==2&&dx==29){
            if(yx%4==0){
                break;
            }
            mx++;
            dx=0;
        }
    }
}
```

```
if(mx==3&&dx==32){
    mx++;
    dx=0;
}
if(mx==4&&dx==31){
    mx++;
    dx=0;
}
if(mx==5&&dx==32){
    mx++;
    dx=0;
}
if(mx==6&&dx==31){
    mx++;
    dx=0;
}
if(mx==7&&dx==32){
    mx++;
    dx=0;
}
if(mx==8&&dx==32){
    mx++;
    dx=0;
}
if(mx==9&&dx==31){
    mx++;
    dx=0;
}
if(mx==10&&dx==32){
    mx++;
    dx=0;
}
if(mx==11&&dx==31){
    mx++;
    dx=0;
}
if(mx==12&&dx==32){
    mx++;
    dx=0;
    yx++;
}
dz++;
if(mz==1&&dz==32){
    mz++;
    dz=0;
}
if(mz==2&&dz==30){
    mz++;
    dz=0;
}
if(mz==2&&dz==29){
    if(yx%4==0){
        break;
    }
}
```

```
        }
        mz++;
        dz=0;
    }
    if(mz==3&&dz==32){
        mz++;
        dz=0;
    }
    if(mz==4&&dz==31){
        mz++;
        dz=0;
    }
    if(mz==5&&dz==32){
        mz++;
        dz=0;
    }
    if(mz==6&&dz==31){
        mz++;
        dz=0;
    }
    if(mz==7&&dz==32){
        mz++;
        dz=0;
    }
    if(mz==8&&dz==32){
        mz++;
        dz=0;
    }
    if(mz==9&&dz==31){
        mz++;
        dz=0;
    }
    if(mz==10&&dz==32){
        mz++;
        dz=0;
    }
    if(mz==11&&dz==31){
        mz++;
        dz=0;
    }
    if(mz==12&&dz==32){
        mz++;
        dz=0;
        yz++;
    }
}

//Transforma ints a string
temp=to_string(yz);
actual=temp;
temp=to_string(mz);
actual+="-";
actual+=temp;
```

```
temp=to_string(dz);
actual+="-";
actual+=temp;

//Edad actual
cout<<"Su edad actual sería: "<<actual;

}

int main(){
    //Fecha de Nacimiento
    string bird;

    //Fecha Actual
    string now;

    //Programa
    cout<<"PROGRAMA PARA CALCULAR SU EDAD ACTUAL A PARTIR DE SU FECHA DE
    NACIMIENTO Y LA FECHA ACTUAL"<<endl;
    cout<<"Ingrese la fecha de su nacimiento (formato YYYY-MM-DD): ";
    getline(cin,bird);
    cout<<"Ingrese la fecha actual: ";
    getline(cin,now);
    actual_Age(bird,now);

    return 0;
}
```

Funcionamiento: Error por alguna razón no bota la edad correctamente.

4. Hacer un programa que desarrolle una función, que genere en pantalla el listado de números primos ubicados entre 1 hasta un numero x (x es ingresado por teclado).

Código:

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

void search_Prim(int a, int* prim){
    //Memoria
    int tempn=1;
    int it=0;
    bool prueba = true;

    //Proceso de búsqueda de primos
    for(int i=0; i<a; i++){
        // Casos especiales
        if (tempn==0||tempn==1||tempn==4){
            prueba=false;
        }
        for (int x=2; x<tempn/2; x++){
            if (tempn%x==0){
                prueba=false;
            }
        }
        if (prueba){
            prim[it]=tempn;
            it++;
        }
        tempn++;
    }
}
```

```
        }
    }
    if(prueba==false){
        tempn++;
    }
    // Si no se pudo dividir por ninguno de los de arriba, sí es primo
    if(prueba==true&&tempn<=a){
        prim[it]=tempn;
        tempn++;
        it++;
    }
    prueba=true;
}

}

int main(){
    //Memoria 1
    int a;
    cout<<"PROGRAMA PARA BUSCAR NUMEROS PRIMOS DESDE 1 HASTA N"<<endl;
    do{
        cout<<"Ingrese un número N: ";
        cin>>a;
        if(a<0){
            cout<<"Error, intentelo de nuevo";
        }
    } while(a<0);
    int* prim = new int[a];
    //Todos los elementos del vector seran 0
    for(int i=0; i<a; i++){
        prim[i]=0;
    }

    search_Prim(a,prim);

    //Mostrar vector
    for(int i=0; i<a; i++){
        if(prim[i]!=0){
            cout<<prim[i]<<" ";
        }
    }

    delete[] prim;

    return 0;
}
```

Funcionamiento:

```
PROGRAMA PARA BUSCAR NUMEROS PRIMOS DESDE 1 HASTA N
Ingrese un número N: 10
2 3 5 7
```

-
5. Desarrollar un programa, utilizando funciones con un tipo de retorno, con las siguientes opciones:
- a. Introducir un valor x entero comprendido entre 0 y 100.
 - b. Validar que sea un número par.
 - c. Sumar todos los números impares desde el 0 hasta el valor de x .

2. Entregables

Al final estudiante deberá:

- 2.1.** Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
- 2.2.** Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe estar en formato PDF.
- 2.3.** El nombre del archivo (comprimido como el documento PDF), será su LAB03_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.
(Ejemplo: LAB03_GRUPO_A_2022123_PEDRO_VASQUEZ).
- 2.4.** Debe remitir el documento ejecutable con el siguiente formato:
LAB03_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO
(Ejemplo: LAB03_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

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