Instituto Politécnico Nacional

Escuela Superior de Ingeniería Mecánica y Eléctrica

Unidad Zacatenco

Departamento de Ingeniería Eléctrica

Programación Orientada a Objetos.

Profesor: Tenorio Huertas José Javier

Alumno: González Gálvez Luis Pablo

“Tarea 4”

“Herencias”

Fecha actual: 17 de octubre del 2020

Fecha de entrega: 17 de octubre del 2020

Índice

Objetivo	3
Introducción	4
Resume (Preliminar)	6
Procedimiento	8
Problema 1	8
Problema 2	17
Problema 3	27
Problema 4	35
Problema 5	44
Conclusiones	51
Bibliografía	51

Objetivo

El objetivo de esta actividad o tarea es hacer que los alumnos entiendan la importancia de las herencias, de las clases padres a las clases hijas, ya que esto puede ser muy útil para reutilizar código, como es el caso de los atributos y de los métodos, y que dado el uso se puedan emplear varios de estos como es el tipo public, private y protected.

Introducción

Herencia

La Herencia es uno de los conceptos fundamentales de la programación orientada a objetos y que permite la reusabilidad de variables y funcionalidades que se han definido en otras clases. Para hablar de herencia se deben introducir los conceptos de clase base y clase derivada. Se conoce como clase base a una clase que va a heredar sus propiedades (variables) y funcionalidades (métodos) a otras clases; por otro lado, se conoce como clase derivada a una clase que se implementa mediante la reutilización de las propiedades y funcionalidades que se heredan de una (o varias) clase base. La herencia contribuye en cierto grado con la escalabilidad de una aplicación ya que cuando se debe modificar o eliminar una variable o un método heredado en todas las clases derivadas, entonces no es necesario que se haga individualmente en cada clase, sino que se hace directamente en la clase base y las clases derivadas simplemente heredan la actualización de esos miembros.

La herencia en C ++ se expresa en la implementación de una clase mediante el uso del operador dos puntos o de ámbito (:) seguido del tipo de herencia y el nombre de la clase base de la que se busca heredar. Vale anotar que el tipo de herencia por defecto en C ++, cuando no se especifica explícitamente ese campo, es la herencia privada. A continuación, se observa la implementación de herencia en C ++:

```
#include<iostream>

using namespace std;

class ClaseBase

{

   protected:

   int unaVar = 0;

   public:

   void unMetodo(void)

   {

      unaVar++;

      cout<<"unaVar = "<<unaVar<<endl;

   }

};
```

```
class ClaseDerivada : public ClaseBase
/* Sintaxis para indicar que ClaseDerivada hereda de ClaseBase */
{
   /* Esta clase implementa los miembros de clase que hereda de ClaseBase */
};
int main()
{
   ClaseDerivada obj1;
   obj1.unMetodo(); /* Acceso a los miembros heredados de ClaseBase */
   obj1.unMetodo();
   return 0;
}
```

Tipos de herencia

Herencia pública: Se refiere a la herencia en la que todos los miembros públicos y protegidos de la clase base conservan esos mismos niveles de acceso respectivamente en las clases derivadas.

```
class ClaseDerivada : public ClaseBase
```

Herencia protegida: Se refiere a la herencia en la que todos los miembros públicos de la clase base adquieren el nivel de acceso protegido en las clases derivadas, mientras que los miembros protegidos conservan su nivel de acceso. Lo anterior indica que una clase derivada puede luego heredar a otra clase los miembros protegidos que heredó de su clase base.

```
class ClaseDerivada : protected ClaseBase
```

Herencia privada: Se refiere a la herencia en la que todos los miembros públicos y protegidos de la clase base adquieren el nivel de acceso privado en las clases derivadas. De ahí se desprende que una clase derivada que haya heredado mediante herencia privada no puede heredar a otras clases los miembros que ha heredado de otras clases.

```
class ClaseDerivada : private ClaseBase
```

Resume Preliminar

El problema #1: En la elaboración del problema se utilizaron cuatro clases, una es la clase base y las otras son las clases derivadas, la clase base se heredó de forma public, en la clase base se declaró un atributo de tipo float, y dos métodos, uno de tipo float y otro de tipo void.

En las clases derivadas, en las dos primeras se hicieron un método tipo float que pide 5 variables de igual modo tipo float, en la ultima clase se declaro un atributo tipo float arreglo, y 5 metodos.

El problema #2: Se usaron cuatro clases, de las cuales una es la clase base y las otras tres son las clases derivadas, la herencia de la clase base es de tipo public, se declararon tres atributos uno de tipo float y dos de tipo string arreglo, y un método para regresar el atributo float.

Dos de las clases derivadas tiene tres métodos, dos de tipo float y uno de tipo void.

La última clase derivada tiene un atributo de tipo float, y tiene cinco métodos de tipo void, dos de estos para calcular el valor de los componentes pedidos.

El problema #3: Se utilizaron cuatro clases, una es la clase base y las otras tres son las derivadas, la clase base se hereda como public; se declaran cuatro atributos de tipo float, y en las clases derivadas en cada una se declaran 2 atributos de tipo float.

En las clases derivadas se crearon seis métodos, cuatro de tipo float y dos de tipo void, y fuera de las clases se declara una función, que pide como valor a los objetos de cada clase, para poder cumplir con la condición previamente pedida en el ejercicio.

El problema #4: En este problema se crearon cinco clases, una clase base y cuatro clases derivadas; en la clase base se declararon doce atributos de tipo const float, la forma en que se heredó la clase base a las derivadas es public.

En las clases derivadas se crean en cada un 9 método, seis de tipo float y tres de tipo void, además de que se crea fuera de las clases una función de tipo void que pide como valor a los objetos y se crea con el propósito de cumplir la condición pedida por el problema.

El problema #5: Se hace uso de tres clases, una es la clase base y las otras dos las derivadas, en la clase base se declaran 7 atributos y cuatro métodos, tres de tipo float y uno de tipo void.

Las clases derivadas tienen 13 métodos, cinco de tipo void y 8 de tipo float, en una de ellas se ejecutan los métodos para mostrarlos en pantalla.

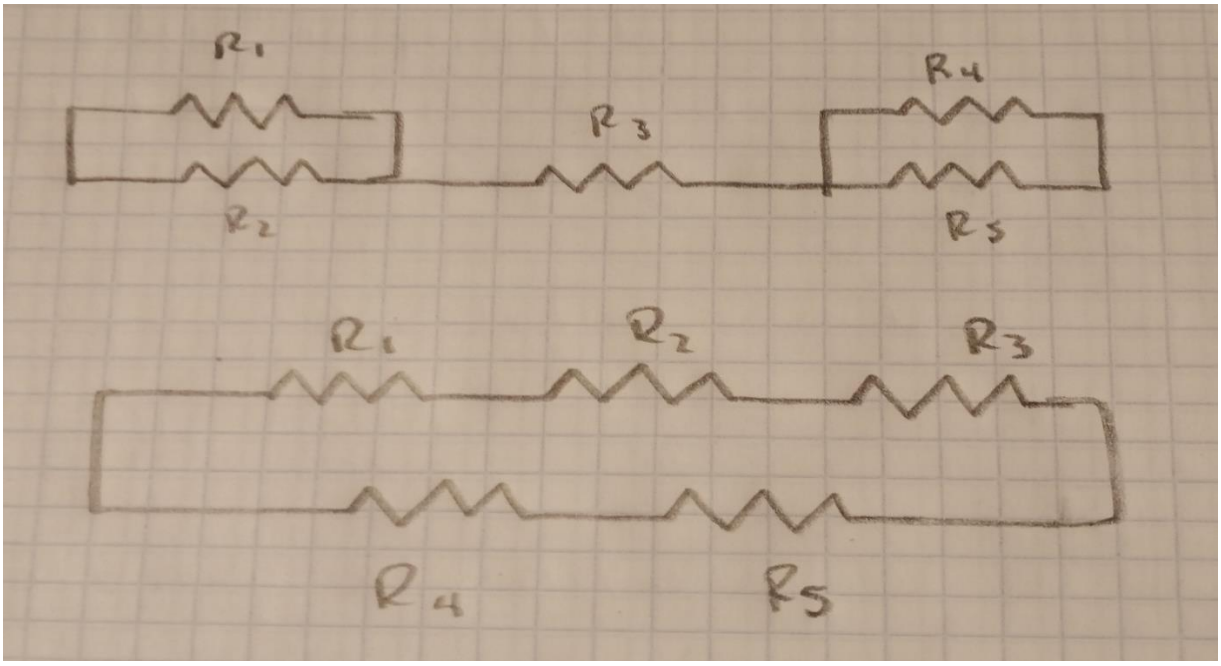
Procedimiento

Problema 1:

El profesor no indico que completáramos el ejemplo visto en la clase de cálculo de resistencias, con las clases paralelo y mixto.

Como el profesor no nos entrego un circuito en el cual basarnos para resolver el circuito en Mixto de resistencias.

Yo realice mi propio circuito se ve en el esquema siguiente:



Syntaxis:

//////////////////////////

//Librerias

//////////////////////////

#include <iostream>

#include <cstdlib>

#include <cmath>

```
#include <ctime>
#include <string.h>
using namespace std;
//////////////////////////
//Definiciones
//////////////////////////
#define tam_max 10
//////////////////////////
//Clases
//////////////////////////
class Circuito{
public:
    float equivalencia;
    float resultado(){
      return equivalencia;
    }
void imprime_serie_para(int n){
    cout<<"Las Resistencias son:"<<endl;
    for(int i=0;  i<n ; i++){
    cout<<"R"<<i+1<<"="<<1<<"ohm ";
    }
    cout<<endl<<endl;
  }
};
```

```
class Serie:public Circuito{
public:
  float suma_resis(float r1,float r2, float r3,float r4, float r5){
    return 1/(r1+r2+r3+r4+r5);
  }
};
class Paralelo:public Circuito{
  public:
  float suma_resis(float r1,float r2, float r3,float r4, float r5){
    return 1/(1/(r1+r2+r3+r4+r5));
  }
};
class Mixto:public Circuito{
  public:
  float r[tam_max];
  void iniciar_resist(int n){
    cout<<"Dame los valores de las Resistencias"<<endl;
    for(int i=0;i<n;i++){
      cout<<"Dame la Resistencia "<<i+1<<": "<<endl;
      cin>>r[i];
    }
  }
  void imprime(int n){
    cout<<"Las Resistencias son:"<<endl;
```

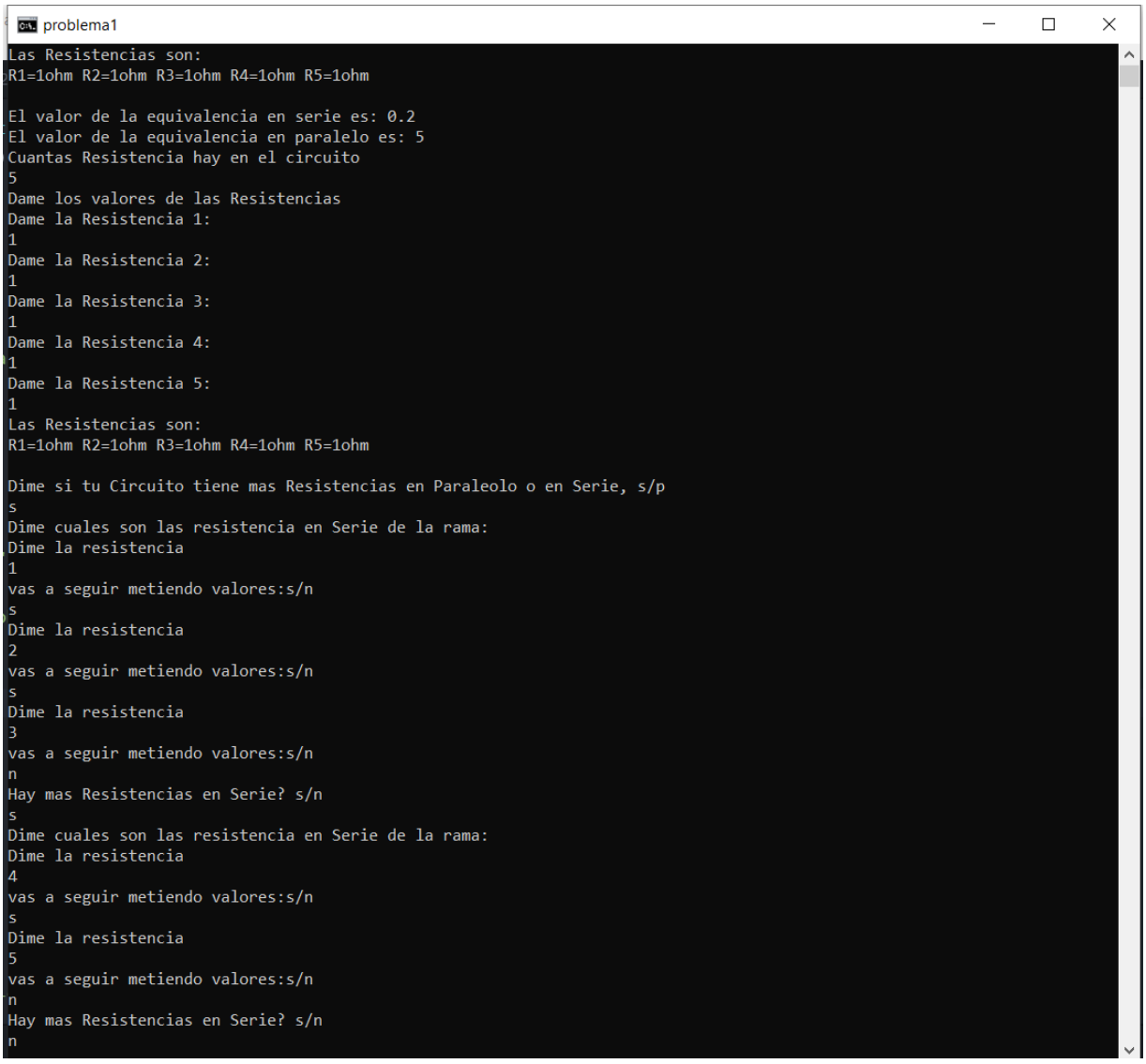
```
  for(int i=0;  i<n ; i++){
  cout<<"R"<<i+1<<"="<<r[i]<<"ohm ";
  }
  cout<<endl<<endl;
}

void pri_serie(int n){
  char a;
  float aux=0;
  int i;
  do{
    cout<<"Dime cuales son las resistencia en Serie de la rama:"<<endl;
    char b='s';
    aux=0;
    for(;b!='n';){
      cout<<"Dime la resistencia"<<endl;
      cin>>i;
      aux+=r[i-1];
      r[i-1]=0;
      cout<<"vas a seguir metiendo valores:s/n"<<endl;
      cin>>b;
      if(b=='n')
      r[i-1]=aux;
    }
```

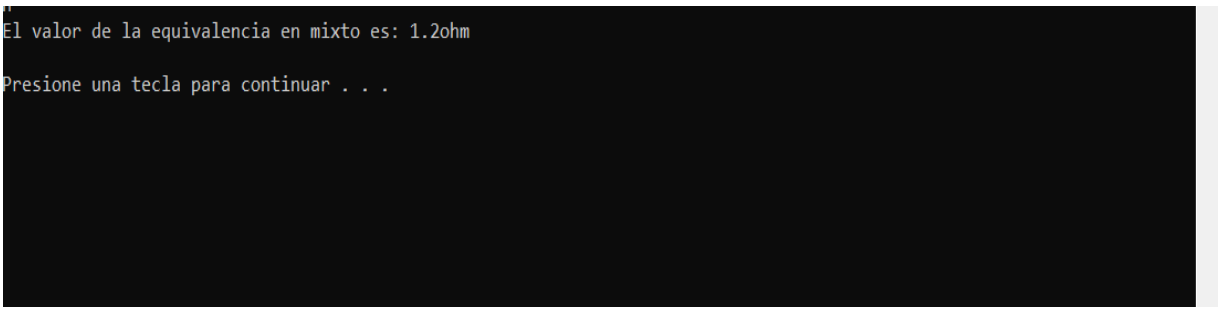
```
    cout<<"Hay mas Resistencias en Serie? s/n"<<endl;
    cin>>a;
  }while(a!='n');
  for(int i=0;i<n;i++){
    if(r[i]!=0){
      r[i]=(1/r[i]);
    }
  }
  equivalencia=0;
  for(int i=0;i<n;i++){
  equivalencia+=(r[i]);
  }
  equivalencia=1/equivalencia;
}
void pri_paralelo(int n){
  char a;
  float aux=0;
  int i;
  do{
    cout<<"Dime cuales son las resistencia en paralelo de la rama:"<<endl;
    char b='s';
    aux=0;
    for(;b!='n';){
      cout<<"Dime la Resistencia"<<endl;
```

```
        cin>>i;
        aux+=1/r[i-1];
        r[i-1]=0;
        cout<<"vas a seguir metiendo valores:s/n"<<endl;
        cin>>b;
        if(b=='n')
        r[i-1]=1/aux;
      }
      cout<<"Hay mas Resistencias en Paralelo? s/n"<<endl;
      cin>>a;
    }while(a!='n');
    equivalencia=0;
    for(int i=0;i<n;i++){
    equivalencia+=(r[i]);
    }
  }
  void suma_resis(int n){
    char b;
      cout<<"Dime si tu Circuito tiene mas Resistencias en Paraleolo o en Serie,
s/p"<<endl;
      cin>>b;
      if(b=='s')
      pri_serie(n);
      else
      pri_paralelo(n);
```

```
  }
};
//////////////////////////
//Funcion Principal
//////////////////////////
int main(){
  Circuito a;
  Serie b;
  Paralelo c;
  Mixto d;
  a.equivalencia=b.suma_resis(1,1,1,1,1);
  cout<<"El valor de la equivalencia en serie es: "<<a.resultado()<<endl;
  a.equivalencia=c.suma_resis(1,1,1,1,1);
  cout<<"El valor de la equivalencia en paralelo es: "<<a.resultado()<<endl;
  int n;
  cout<<"Cuantas Resistencia hay en el circuito"<<endl;
  cin>>n;
  d.iniciar_resist(n);
  d.imprime(n);
  d.suma_resis(n);
  cout<<"El valor de la equivalencia en mixto es: "<<d.resultado()<<”ohm”<<endl;
}
```

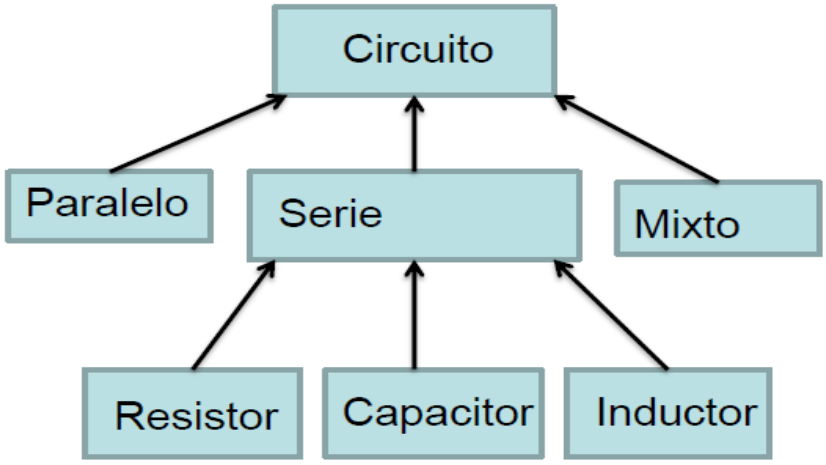

problema1
Las Resistencias son:
R1=1ohm R2=1ohm R3=1ohm R4=1ohm R5=1ohm
El valor de la equivalencia en serie es: 0.2
El valor de la equivalencia en paralelo es: 5
Cuantas Resistencia hay en el circuito
5
Dame los valores de las Resistencias
Dame la Resistencia 1:
1
Dame la Resistencia 2:
1
Dame la Resistencia 3:
1
Dame la Resistencia 4:
1
Dame la Resistencia 5:
1
Las Resistencias son:
R1=1ohm R2=1ohm R3=1ohm R4=1ohm R5=1ohm
Dime si tu Circuito tiene mas Resistencias en Paraleolo o en Serie, s/p
p
Dime cuales son las resistencia en paralelo de la rama:
Dime la Resistencia
1
vas a seguir metiendo valores:s/n
s
Dime la Resistencia
2
vas a seguir metiendo valores:s/n
n
Hay mas Resistencias en Paralelo? s/n
s
Dime cuales son las resistencia en paralelo de la rama:
Dime la Resistencia
4
vas a seguir metiendo valores:s/n
s
Dime la Resistencia
5
vas a seguir metiendo valores:s/n
n
Hay mas Resistencias en Paralelo? s/n
n
El valor de la equivalencia en mixto es: 2ohm
Presione una tecla para continuar . . .


problema1
Las Resistencias son:
R1=1ohm R2=1ohm R3=1ohm R4=1ohm R5=1ohm
El valor de la equivalencia en serie es: 0.2
El valor de la equivalencia en paralelo es: 5
Cuantas Resistencia hay en el circuito
5
Dame los valores de las Resistencias
Dame la Resistencia 1:
1
Dame la Resistencia 2:
1
Dame la Resistencia 3:
1
Dame la Resistencia 4:
1
Dame la Resistencia 5:
1
Las Resistencias son:
R1=1ohm R2=1ohm R3=1ohm R4=1ohm R5=1ohm
Dime si tu Circuito tiene mas Resistencias en Paraleolo o en Serie, s/p
s
Dime cuales son las resistencia en Serie de la rama:
Dime la resistencia
1
vas a seguir metiendo valores:s/n
s
Dime la resistencia
2
vas a seguir metiendo valores:s/n
s
Dime la resistencia
3
vas a seguir metiendo valores:s/n
n
Hay mas Resistencias en Serie? s/n
s
Dime cuales son las resistencia en Serie de la rama:
Dime la resistencia
4
vas a seguir metiendo valores:s/n
s
Dime la resistencia
5
vas a seguir metiendo valores:s/n
n
Hay mas Resistencias en Serie? s/n
n


El valor de la equivalencia en mixto es: 1.2ohm
Presione una tecla para continuar . . .

Problema 2:

Elabore un algoritmo que, haciendo uso de herencias, calcule el valor equivalente de cinco elementos (del mismo tipo) de un circuito. Ver el siguiente esquema:



Las fórmulas por utilizar son las siguientes:

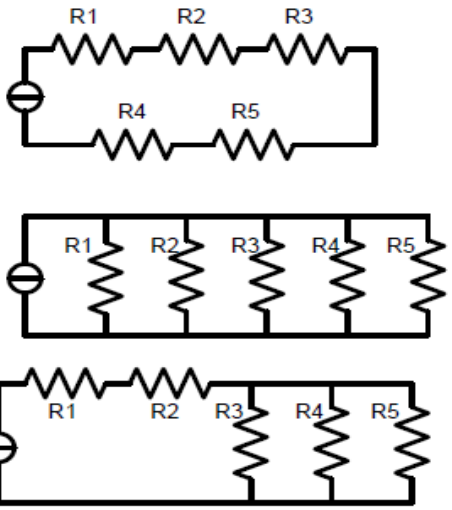
• Para las resistencias en serie Req=R1+R2+R3+R4+R5

• Para las resistencias en paralelo Req=1/(1/R1+ 1/R2+ 1/R3+ 1/R4+ 1/R5)

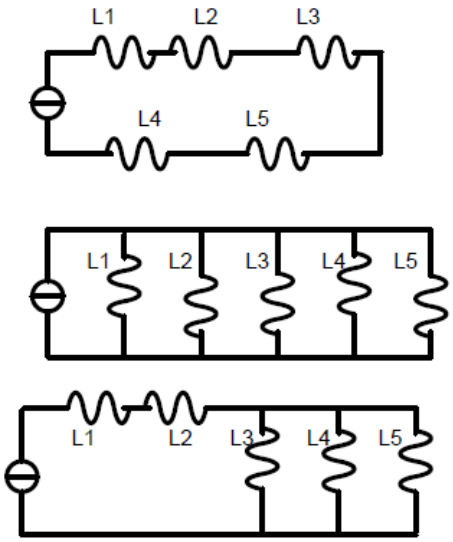
• Para las resistencias en mixto: Se obtienen dos equivalentes, en el primer equivalente los resistores 3, 4 y 5 se calculan Req2=1/(1/R3+1/R4+ 1/R5, y posteriormente se calculan en serie, es decir Req2=R1+R2+Req2.

Nota:

Los valores de las resistencias se dan en ohms: Por ejemplo, R1=3 ohms.



Para las inductancias, las ecuaciones son las mismas, que, en los resistores, y sus valores se dan, generalmente, en mili henrios (mH), por ejemplo L1=6 mH.

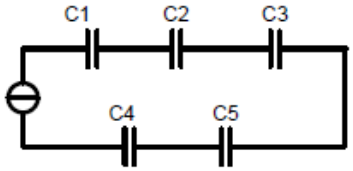


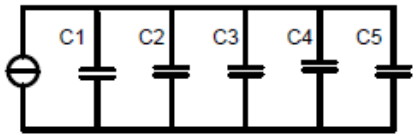
Las fórmulas por utilizar, para la capacitancia, son las siguientes:

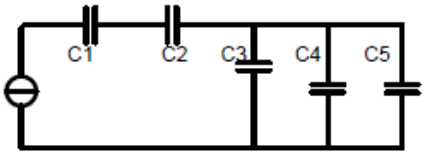
• Para las capacitancia en serie: Ceq=1/(1/R1+ 1/R2+ 1/R3+ 1/R4+ 1/R5).

• Para las capacitancias en paralelo: Ceq=R1+R2+R3+R4+R5.

• Para las capacitancias en mixto: Se obtienen dos equivalentes, en el primer equivalente los capacitores 3, 4 y 5 se calculan Ceq1=R3+R4+R5, y posteriormente se calculan Req2=1/(1/C1+1/C2+1/Ceq2).







Syntaxis:

```
//////////////////////
//Librerias
//////////////////////
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <string.h>
using namespace std;
//////////////////////
//Definiciones
//////////////////////
#define tam_max 10
//////////////////////
//Clases
//////////////////////
class Circuito{
public:
    float equivalencia;
    string s[tam_max]{"resistencia","inductancias"};
```

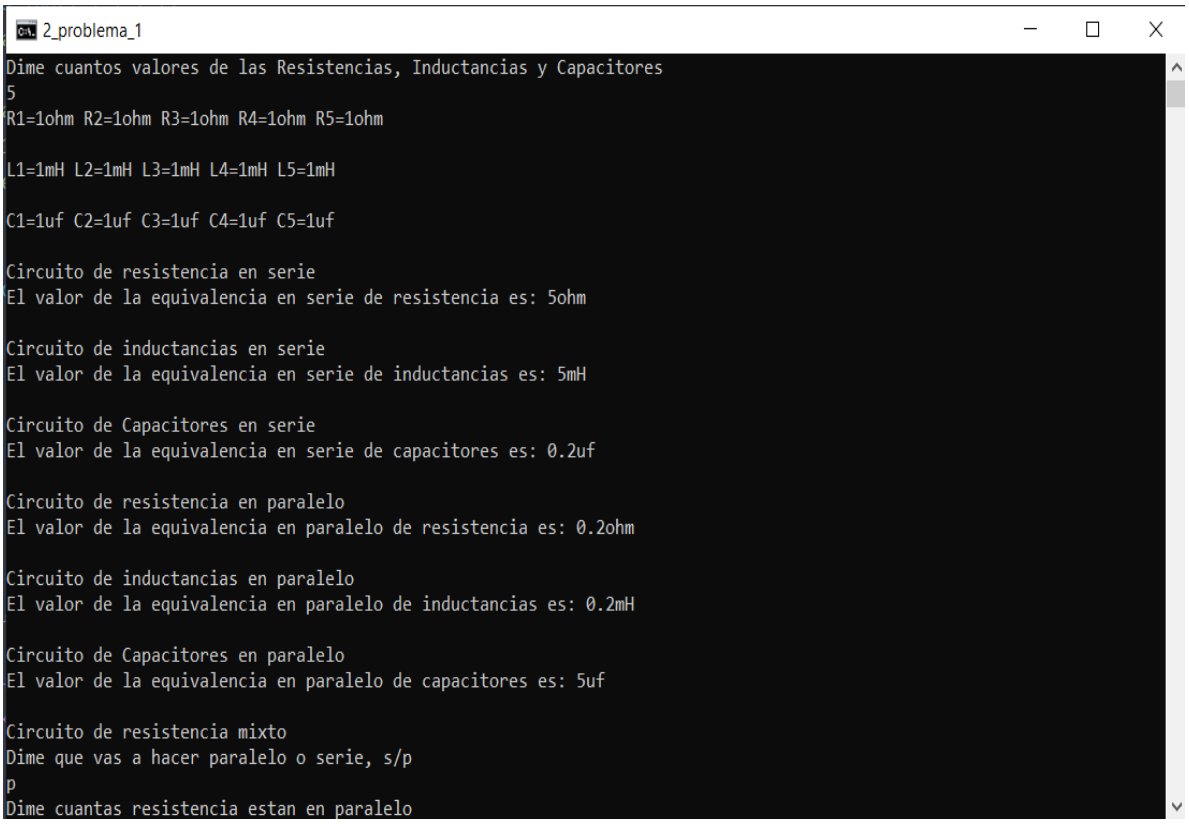
```
    string u[tam_max]{"ohm", "mH", "uf"};
    float resultado(){
      return equivalencia;
    }
};
class Serie:public Circuito{
public:
  float suma_res_ind(float r1,float r2, float r3,float r4, float r5){
    return (r1+r2+r3+r4+r5);
  }
  float suma_cap(float r1,float r2, float r3,float r4, float r5){
    return (1/(r1+r2+r3+r4+r5));
  }
  void todo_serie(){
    for(int i=0;i<2;i++){
      cout<<"Circuito de "<<s[i]<<" en serie"<<endl;
      equivalencia=suma_res_ind(1,1,1,1,1);
      cout<<"El valor de la equivalencia en serie de "<<s[i]<<" es:
"<<resultado()<<u[i]<<endl<<endl;
    }
    cout<<"Circuito de Capacitores en serie"<<endl;
    equivalencia=suma_cap(1,1,1,1,1);
    cout<<"El valor de la equivalencia en serie de capacitores es:
"<<resultado()<<u[2]<<endl<<endl;
  }
};
class Paralelo:public Circuito{
  public:
  float suma_res_ind(float r1,float r2, float r3,float r4, float r5){
    return (1/(r1+r2+r3+r4+r5));
  }
  float suma_cap(float r1,float r2, float r3,float r4, float r5){
    return (r1+r2+r3+r4+r5);
```

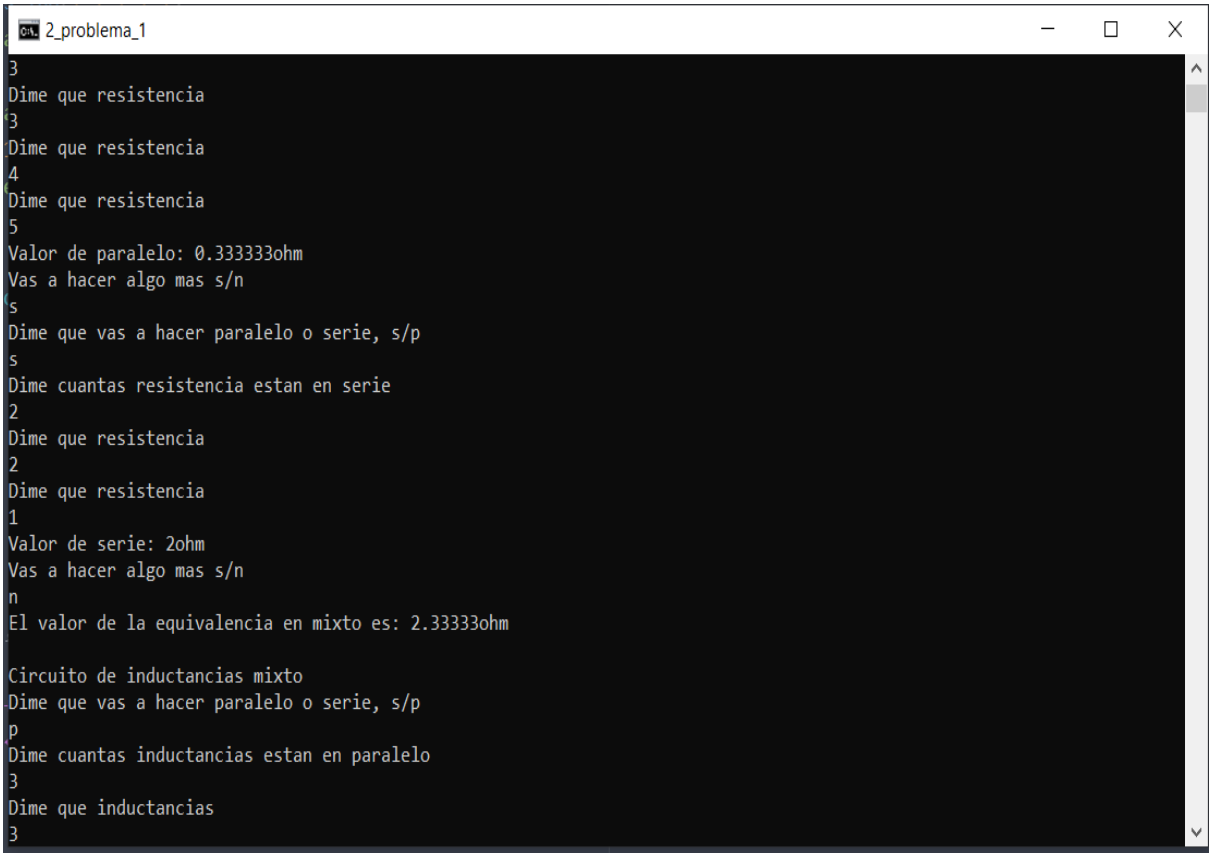
```
  }
  void todo_paralelo(){
    for(int i=0;i<2;i++){
      cout<<"Circuito de "<<s[i]<<" en paralelo"<<endl;
      equivalencia=suma_res_ind(1,1,1,1,1);
      cout<<"El valor de la equivalencia en paralelo de "<<s[i]<<" es:
"<<resultado()<<u[i]<<endl<<endl;
    }
    cout<<"Circuito de Capacitores en paralelo"<<endl;
    equivalencia=suma_cap(1,1,1,1,1);
    cout<<"El valor de la equivalencia en paralelo de capacitores es:
"<<resultado()<<u[2]<<endl<<endl;
  }
};
class Mixto:public Circuito{
  public:
  float r[tam_max];

void iniciar(int n){
    for(int i=0;i<n;i++)
      r[i]=1;
}
void imprime(int n){
  char c[tam_max]={'R','L','C'};
  for(int j=0;j<3;j++){
    for(int i=0;  i<n ; i++){
      cout<<c[j]<<i+1<<"="<<r[i]<<u[j]<<" ";
    }
    cout<<endl<<endl;
  }
}
void mix_res_ind(int l){
  float aux_ser=0;
```

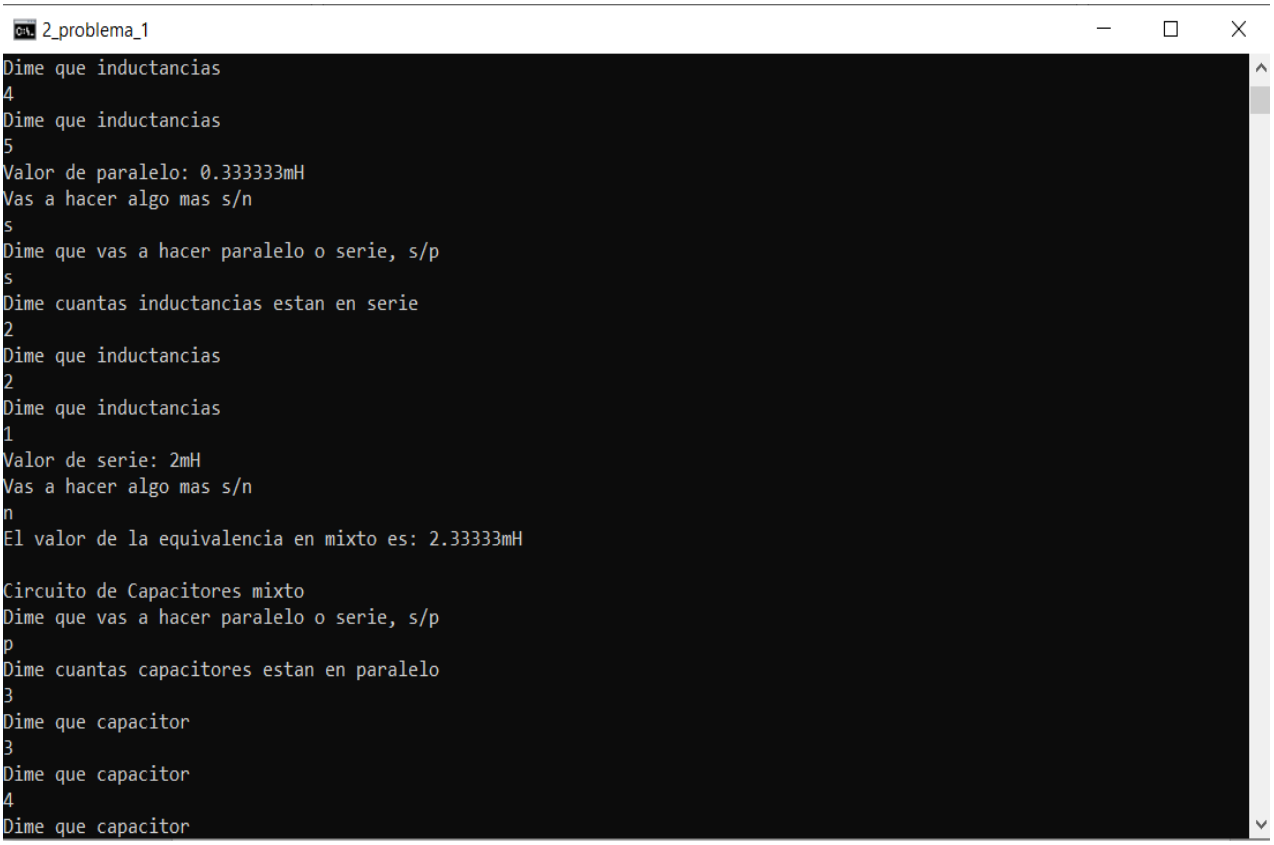
```
  float aux_para=0;
  int i,n;
  char a;
    do{
    cout<<"Dime que vas a hacer paralelo o serie, s/p"<<endl;
    cin>>a;
    if(a=='p'){
      cout<<"Dime cuantas "<<s[l]<<" estan en paralelo"<<endl;
      cin>>n;
      for(int j=0;j<n;j++){
        cout<<"Dime que "<<s[l]<<endl;
        cin>>i;
        aux_para+=1/r[i-1];
      }
      aux_para=1/aux_para;
      cout<<"Valor de paralelo: "<<aux_para<<u[l]<<endl;
    }
    else{
      cout<<"Dime cuantas "<<s[l]<<" estan en serie"<<endl;
      cin>>n;
      for(int j=0;j<n;j++){
        cout<<"Dime que "<<s[l]<<endl;
        cin>>i;
        aux_ser+=r[i-1];
      }
      cout<<"Valor de serie: "<<aux_ser<<u[l]<<endl;
    }
    cout<<"Vas a hacer algo mas s/n"<<endl;
    cin>>a;
    }while(a!='n');
    equivalencia=aux_ser+aux_para;
    cout<<"El valor de la equivalencia en mixto es:
"<<resultado()<<u[l]<<endl<<endl;
```

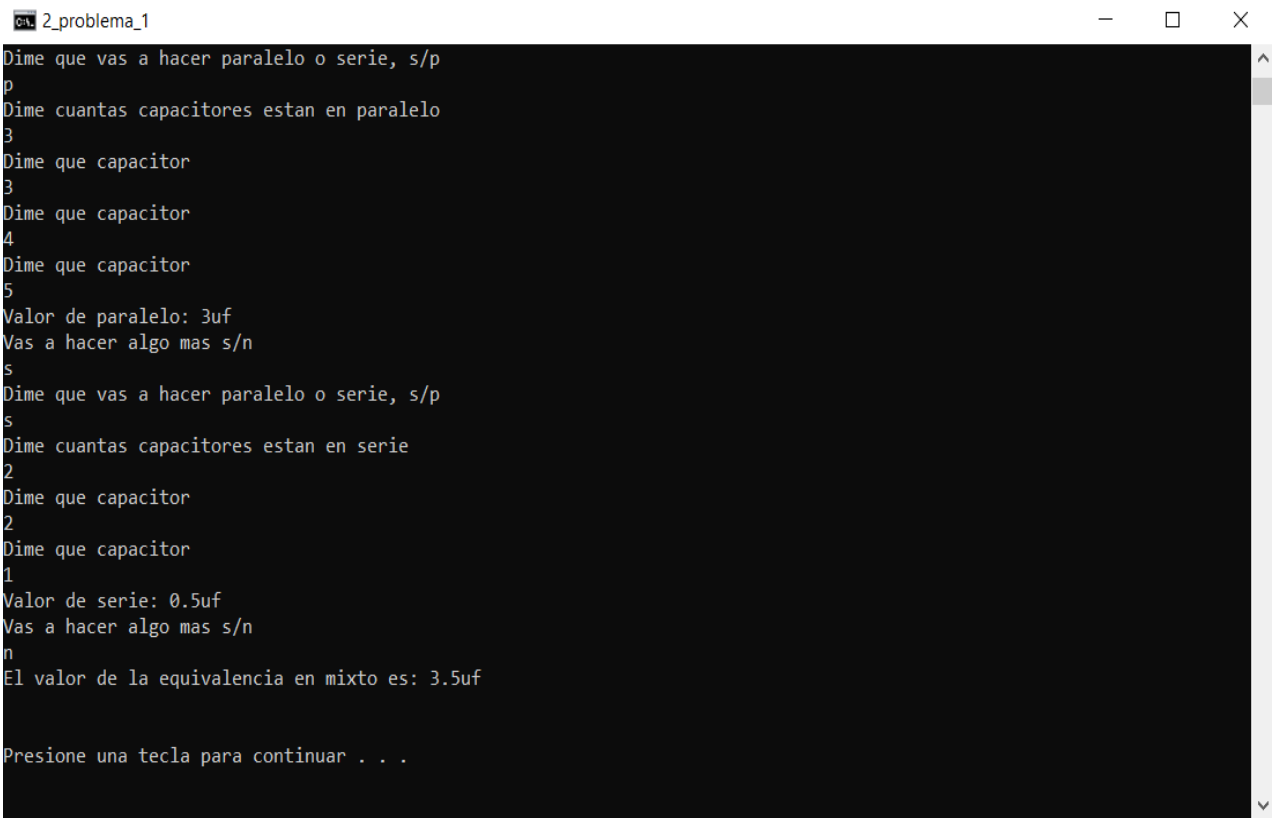
```
}
void mix_cap(){
  float aux_ser=0;
  float aux_para=0;
  int i,n;
  char a;
    do{
    cout<<"Dime que vas a hacer paralelo o serie, s/p"<<endl;
    cin>>a;
    if(a=='p'){
      cout<<"Dime cuantas capacitores estan en paralelo"<<endl;
      cin>>n;
      for(int j=0;j<n;j++){
        cout<<"Dime que capacitor"<<endl;
        cin>>i;
        aux_para+=r[i-1];
      }
      cout<<"Valor de paralelo: "<<aux_para<<u[2]<<endl;
    }
    else{
      cout<<"Dime cuantas capacitores estan en serie"<<endl;
      cin>>n;
      for(int j=0;j<n;j++){
        cout<<"Dime que capacitor"<<endl;
        cin>>i;
        aux_ser+=1/r[i-1];
      }
      aux_ser=1/aux_ser;
      cout<<"Valor de serie: "<<aux_ser<<u[2]<<endl;
    }
    cout<<"Vas a hacer algo mas s/n"<<endl;
    cin>>a;
    }while(a!='n');
```

```
      equivalencia=aux_ser+aux_para;
      cout<<"El valor de la equivalencia en mixto es:
"<<resultado()<<u[2]<<endl<<endl;
    }
  void todo_mix(){
    for(int i=0;i<2;i++){
      cout<<"Circuito de "<<s[i]<<" mixto"<<endl;
      mix_res_ind(i);
    }
    cout<<"Circuito de Capacitores mixto"<<endl;
    mix_cap();
  }
};
//////////////////////////
//Funcion Principal
//////////////////////////
int main(){
  Circuito a;
  Serie b;
  Paralelo c;
  Mixto d;
  int n;
  cout<<"Dime cuantos valores de las Resistencias, Inductancias y
Capacitores"<<endl;
  cin>>n;
  d.iniciar(n);
  d.imprime(n);
  b.todo_serie();
  c.todo_paralelo();
  d.todo_mix();
}
```


2_problema_1
Dime cuantos valores de las Resistencias, Inductancias y Capacitores
5
R1=1ohm R2=1ohm R3=1ohm R4=1ohm R5=1ohm
L1=1mH L2=1mH L3=1mH L4=1mH L5=1mH
C1=1uf C2=1uf C3=1uf C4=1uf C5=1uf
Circuito de resistencia en serie
El valor de la equivalencia en serie de resistencia es: 5ohm
Circuito de inductancias en serie
El valor de la equivalencia en serie de inductancias es: 5mH
Circuito de Capacitores en serie
El valor de la equivalencia en serie de capacitores es: 0.2uf
Circuito de resistencia en paralelo
El valor de la equivalencia en paralelo de resistencia es: 0.2ohm
Circuito de inductancias en paralelo
El valor de la equivalencia en paralelo de inductancias es: 0.2mH
Circuito de Capacitores en paralelo
El valor de la equivalencia en paralelo de capacitores es: 5uf
Circuito de resistencia mixto
Dime que vas a hacer paralelo o serie, s/p
p
Dime cuantas resistencia estan en paralelo

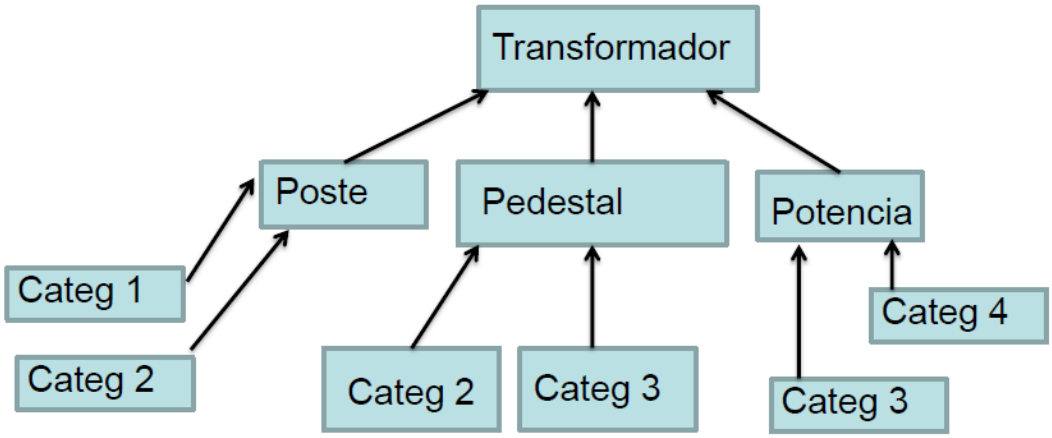

2_problema_1
3
Dime que resistencia
3
Dime que resistencia
4
Dime que resistencia
5
Valor de paralelo: 0.333333ohm
Vas a hacer algo mas s/n
s
Dime que vas a hacer paralelo o serie, s/p
s
Dime cuantas resistencia estan en serie
2
Dime que resistencia
2
Dime que resistencia
1
Valor de serie: 2ohm
Vas a hacer algo mas s/n
n
El valor de la equivalencia en mixto es: 2.33333ohm
Circuito de inductancias mixto
Dime que vas a hacer paralelo o serie, s/p
p
Dime cuantas inductancias estan en paralelo
3
Dime que inductancias
3


2_problema_1
Dime que inductancias
4
Dime que inductancias
5
Valor de paralelo: 0.333333mH
Vas a hacer algo mas s/n
s
Dime que vas a hacer paralelo o serie, s/p
s
Dime cuantas inductancias estan en serie
2
Dime que inductancias
2
Dime que inductancias
1
Valor de serie: 2mH
Vas a hacer algo mas s/n
n
El valor de la equivalencia en mixto es: 2.33333mH
Circuito de Capacitores mixto
Dime que vas a hacer paralelo o serie, s/p
p
Dime cuantas capacitores estan en paralelo
3
Dime que capacitor
3
Dime que capacitor
4
Dime que capacitor


2_problema_1
Dime que vas a hacer paralelo o serie, s/p
p
Dime cuantas capacitores estan en paralelo
3
Dime que capacitor
3
Dime que capacitor
4
Dime que capacitor
5
Valor de paralelo: 3uf
Vas a hacer algo mas s/n
s
Dime que vas a hacer paralelo o serie, s/p
s
Dime cuantas capacitores estan en serie
2
Dime que capacitor
2
Dime que capacitor
1
Valor de serie: 0.5uf
Vas a hacer algo mas s/n
n
El valor de la equivalencia en mixto es: 3.5uf
Presione una tecla para continuar . . .

Problema 3:

Elabore un algoritmo que, haciendo uso de herencias, calcule la corriente primaria y secundaria de un transformador, de acuerdo a su potencia en kVA . Ver el siguiente esquema:



No olvidar que la categoría de cada transformador esta en función de su capacidad, ver tabla siguiente:

Categorías	Potencia(kVAs)
1	15 – 500
2	501 – 5000
3	5001 – 30000
4	Mayor a 30000

Y que la corriente primaria y secundaria se calcula con:

Ip= kVA /Raiz (3) * kVp y Is= kVA /Raiz (3) *kVs

Tomando en cuenta que para el tipo poste y pedestal, el Vp =23 kV,

Vs=0.44 kV, y para el transformador tipo potencia el Vp =85 kV y Vs=23

kV

Syntaxis:

//////////////////////////

//Librerias

//////////////////////////

#include <iostream>

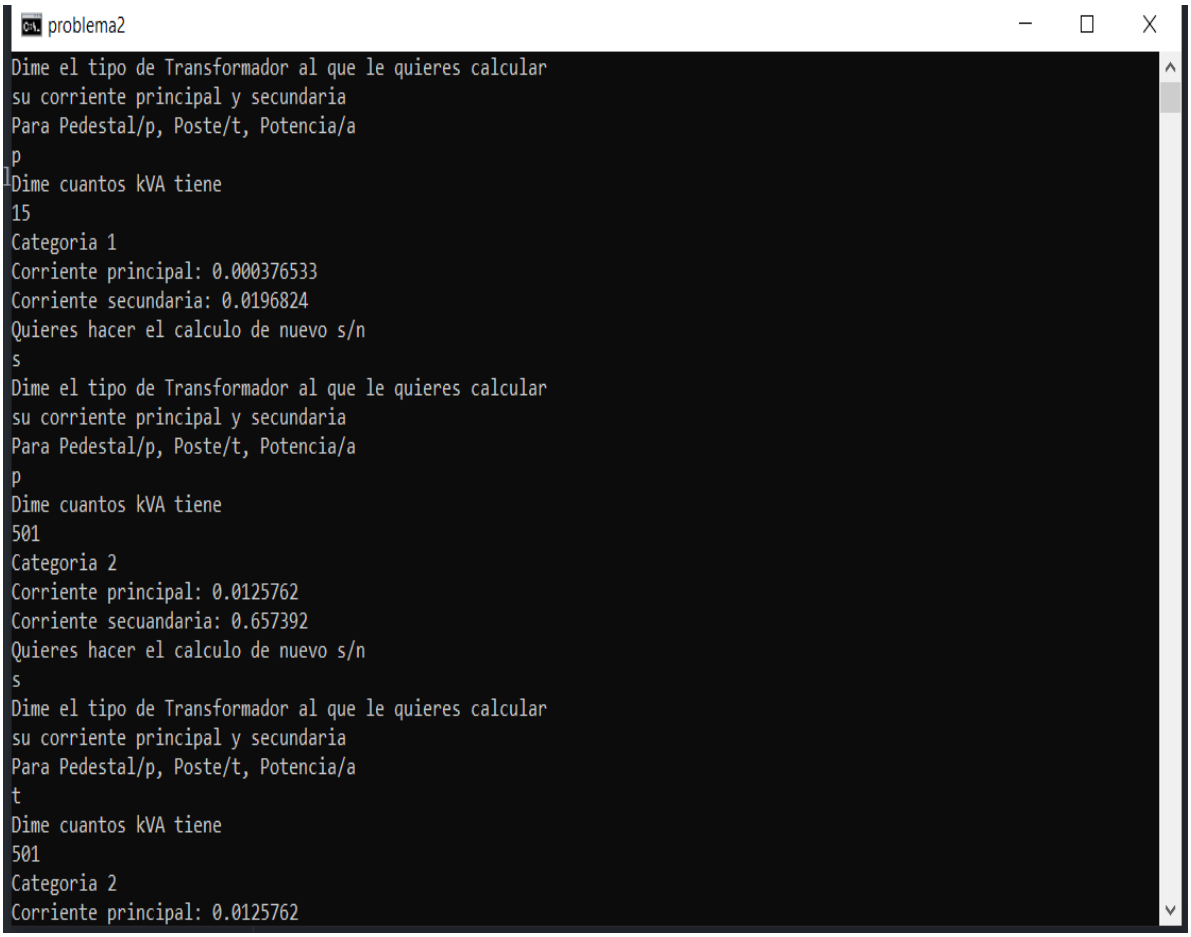
```
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <string.h>
using namespace std;
//////////////////////////
//Definiciones
//////////////////////////
#define tam_max 10
//////////////////////////
//Clases
//////////////////////////
class Transformador{
public:
  float vp1=23;
  float vp2=85;
  float vs1=0.44;
  float vs2=23;
};
class Poste:public Transformador{
  float lp;
  float ls;
public:
  void co_pri2(float kVA){
    lp=kVA/(sqrt(3)*1000*vp1);
  }
  void co_sec2(float kVA){
    ls=kVA/(sqrt(3)*1000*vs1);
  }
  float co_pri(float kVA){
      co_pri2(kVA);
      return get_prin();
  }
```

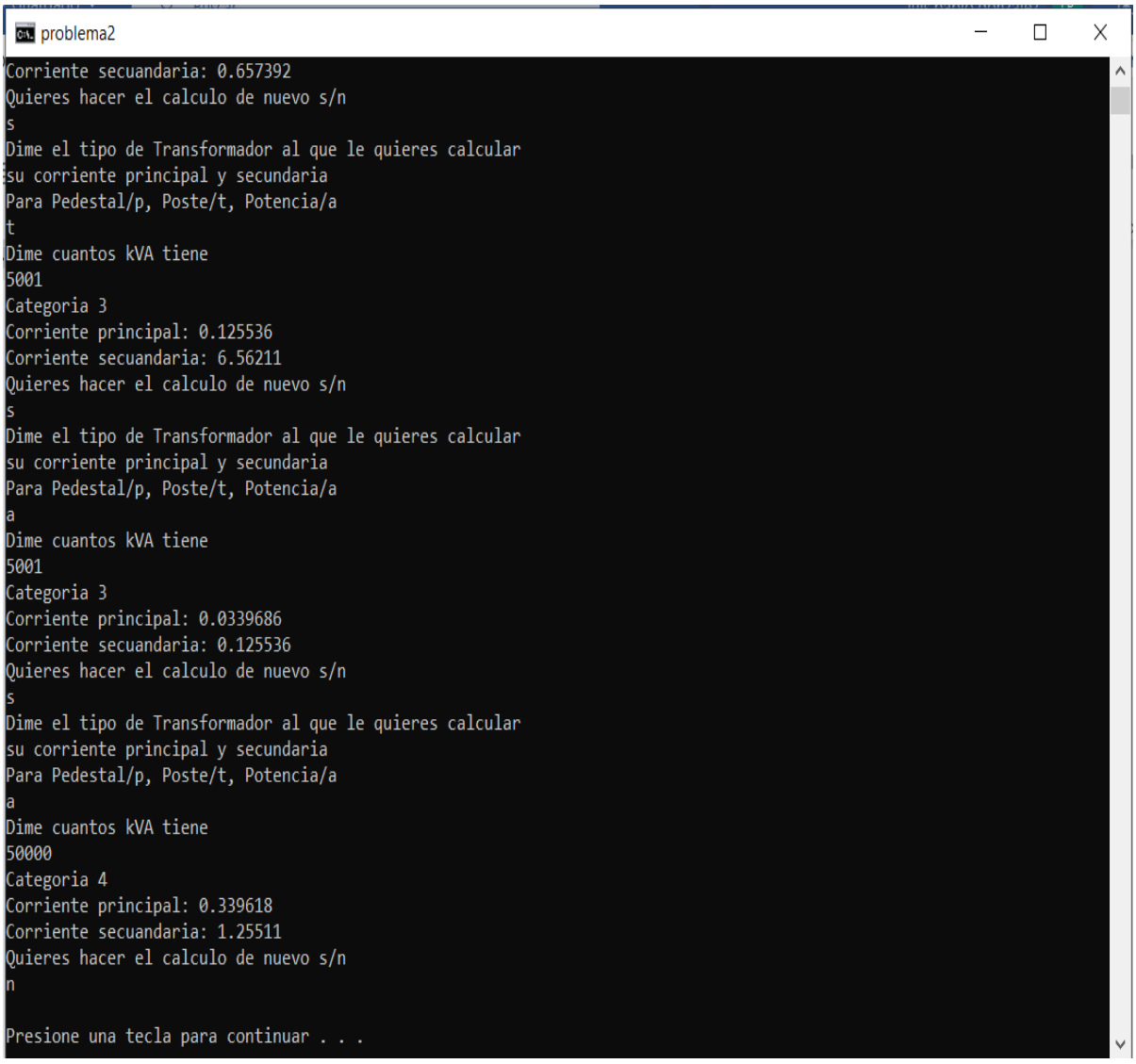
```
  float co_sec(float kVA){
      co_sec2(kVA);
      return get_sec();
  }
  float get_prin(){
      return lp;
  }
  float get_sec(){
      return ls;
  }
};
class Pedestal:public Transformador{
  float lp;
  float ls;
public:
  void co_pri2(float kVA){
    lp=kVA/(sqrt(3)*1000*vp1);
  }
  void co_sec2(float kVA){
    ls=kVA/(sqrt(3)*1000*vs1);
  }
  float co_pri(float kVA){
      co_pri2(kVA);
      return get_prin();
  }
  float co_sec(float kVA){
      co_sec2(kVA);
      return get_sec();
  }
  float get_prin(){
      return lp;
  }
  float get_sec(){
```

```
      return ls;
  }
};
class Potencia:public Transformador{
  float lp;
  float ls;
public:
  void co_pri2(float kVA){
    lp=kVA/(sqrt(3)*1000*vp2);
  }
  //auto completador de codigo, generedor de seters y geters atom
  void co_sec2(float kVA){
    ls=kVA/(sqrt(3)*1000*vs2);
  }
  float co_pri(float kVA){
      co_pri2(kVA);
      return get_prin();
  }
  float get_prin(){
      return lp;
  }
  float get_sec(){
      return ls;
  }
  float co_sec(float kVA){
      co_sec2(kVA);
      return get_sec();
  }
};
void todo(Poste b,Pedestal c, Potencia d);
//////////////////////////
//Funcion Principal
//////////////////////////
```

```
int main(){
  Transformador a;
  Poste b;
  Pedestal c;
  Potencia d;
  todo(b,c,d);
}
void todo(Poste b,Pedestal c, Potencia d){
  char n;
  int f;
  char s;
  do{
  cout<<"Dime el tipo de Transformador al que le quieres calcular"<<endl;
  cout<<"su corriente principal y secundaria"<<endl;
  cout<<"Para Pedestal/p, Poste/t, Potencia/a"<<endl;
  cin>>n;
  if(n=='p'||n=='t'){
    cout<<"Dime cuantos kVA tiene"<<endl;
    cin>>f;
    if(f>=15&&f<=500){
      cout<<"Categoria 1"<<endl;
      cout<<"Corriente principal: "<<b.co_pri(f)<<endl;
      cout<<"Corriente secundaria: "<<b.co_sec(f)<<endl;
    }
    else if(f>=501&&f<=5000){
      if(n=='p'){
        cout<<"Categoria 2"<<endl;
        cout<<"Corriente principal: "<<b.co_pri(f)<<endl;
        cout<<"Corriente secuandaria: "<<b.co_sec(f)<<endl;
      }
      else{
        cout<<"Categoria 2"<<endl;
        cout<<"Corriente principal: "<<c.co_pri(f)<<endl;
```

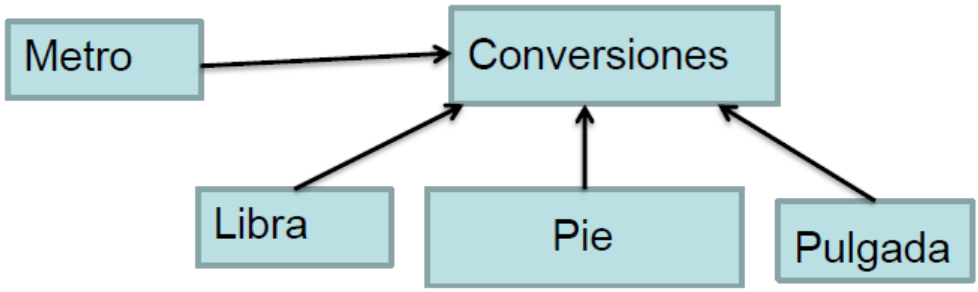
```
        cout<<"Corriente secuandaria: "<<c.co_sec(f)<<endl;
      }
    }
    else if(f>=5001&&f<=30000){
      cout<<"Categoria 3"<<endl;
      cout<<"Corriente principal: "<<c.co_pri(f)<<endl;
      cout<<"Corriente secuandaria: "<<c.co_sec(f)<<endl;
    }
  }
  else{
    cout<<"Dime cuantos kVA tiene"<<endl;
    cin>>f;
    if(f>=5001&&f<=30000){
      cout<<"Categoria 3"<<endl;
      cout<<"Corriente principal: "<<d.co_pri(f)<<endl;
      cout<<"Corriente secuandaria: "<<d.co_sec(f)<<endl;
    }
    else{
      cout<<"Categoria 4"<<endl;
      cout<<"Corriente principal: "<<d.co_pri(f)<<endl;
      cout<<"Corriente secuandaria: "<<d.co_sec(f)<<endl;
    }
  }
cout<<"Quieres hacer el calculo de nuevo s/n"<<endl;
cin>>s;
}while(s!='n');
}
```


problema2
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
p
Dime cuantos kVA tiene
15
Categoria 1
Corriente principal: 0.000376533
Corriente secundaria: 0.0196824
Quieres hacer el calculo de nuevo s/n
s
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
p
Dime cuantos kVA tiene
501
Categoria 2
Corriente principal: 0.0125762
Corriente secuandaria: 0.657392
Quieres hacer el calculo de nuevo s/n
s
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
t
Dime cuantos kVA tiene
501
Categoria 2
Corriente principal: 0.0125762


problema2
Corriente secuandaria: 0.657392
Quieres hacer el calculo de nuevo s/n
s
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
t
Dime cuantos kVA tiene
5001
Categoria 3
Corriente principal: 0.125536
Corriente secuandaria: 6.56211
Quieres hacer el calculo de nuevo s/n
s
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
a
Dime cuantos kVA tiene
5001
Categoria 3
Corriente principal: 0.0339686
Corriente secuandaria: 0.125536
Quieres hacer el calculo de nuevo s/n
s
Dime el tipo de Transformador al que le quieres calcular
su corriente principal y secundaria
Para Pedestal/p, Poste/t, Potencia/a
a
Dime cuantos kVA tiene
50000
Categoria 4
Corriente principal: 0.339618
Corriente secuandaria: 1.25511
Quieres hacer el calculo de nuevo s/n
n
Presione una tecla para continuar . . .

Problema 4:

Elabore un algoritmo que, haciendo uso de herencias, y la entrada de un valor de longitud, calcule las conversiones requeridas. Ver el siguiente esquema:



Si la entrada es en Metros, se deberá convertir a Libras, Pies y Pulgadas.

Si la entrada es en Libras, se deberá convertir a Metros, Pies y Pulgadas.

Si la entrada es en Pies, se deberá convertir a Metros, Libras y Pulgadas.

Si la entrada es en Pulgadas, se deberá convertir a Metros, Libras y Pies.

Syntaxis:

```
//////////////////////////
//Librerias
//////////////////////////
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <string.h>
using namespace std;
//////////////////////////
//Definiciones
//////////////////////////
#define tam_max 10
//////////////////////////
//Clases
//////////////////////////
class Conversion{
```

```
public:
  const float m_pi=3.28;
  const float m_pul=39.37;
  const float m_li=0.7376;
  const float li_pi=0.083333;
  const float li_pul=2.036;
  const float li_m=0.01152;
  const float pi_m=0.3048;
  const float pi_pul=12;
  const float pi_li=0.083333;
  const float pul_pi=0.0833333;
  const float pul_m=0.0254;
  const float pul_li=2.036;
};
class Metros:Conversion{
  float pul;
  float libra;
  float pies;
public:
  void metros_pies(float m){
    pies=m*m_pi;
  }
  void metros_pul(float m){
    pul=m*m_pul;
  }
  void metros_libras(float m){
    libra=m*m_li;
  }
  float met_pie(float m){
    metros_pies(m);
    return ret_met_pie();
  }
  float met_pul(float m){
```

```
    metros_pul(m);
    return ret_met_pul();
  }
  float met_lib(float m){
    metros_libras(m);
    return ret_met_lib();
  }
  float ret_met_pul(){
    return pul;
  }
  float ret_met_lib(){
    return libra;
  }
  float ret_met_pie(){
    return pies;
  }
};
class Pulgadas:Conversion{
  float m;
  float pie;
  float li;
public:
  void pul_met(int pul){
    m=pul*pul_m;
  }
  void pul_pie(int pul){
    pie=pul*pul_pi;
  }
  void pul_libras(int pul){
    li=pul*pul_li;
  }
  float pulg_met(float m){
    pul_met(m);
```

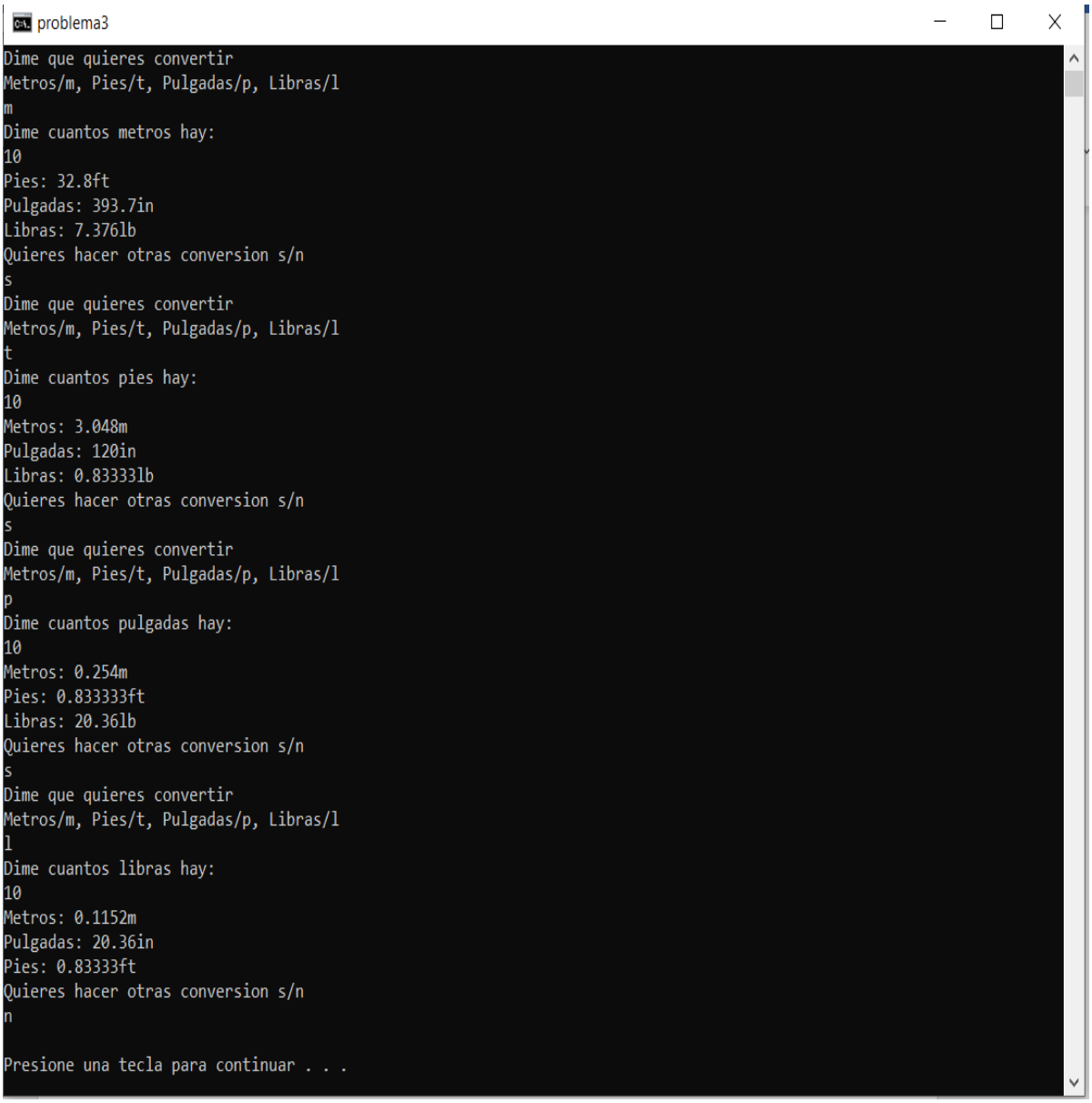
```
    return ret_pul_met();
  }
  float pulg_pie(float m){
    pul_pie(m);
    return ret_pul_pie();
  }
  float pulg_lib(float m){
    pul_libras(m);
    return ret_pul_lib();
  }
  float ret_pul_lib(){
    return li;
  }
  float ret_pul_met(){
    return m;
  }
  float ret_pul_pie(){
    return pie;
  }
};
class Pies:Conversion{
  float m;
  float pul;
  float li;
  public:
  void pie_met(float pi){
    m=pi*pi_m;
  }
  void pie_pul(float pi){
    pul=pi*pi_pul;
  }
  void pie_libra(float pi){
    li=pi*pi_li;
```

```
  }
  float pies_met(float m){
    pie_met(m);
    return ret_pie_met();
  }
  float pies_pul(float m){
    pie_pul(m);
    return ret_pie_pul();
  }
  float pies_lib(float m){
    pie_libra(m);
    return ret_pie_lib();
  }
  float ret_pie_lib(){
    return li;
  }
  float ret_pie_met(){
    return m;
  }
  float ret_pie_pul(){
    return pul;
  }
};
class Libras:Conversion{
  float pul;
  float m;
  float pie;
  public:
  void libra_met(float lib){
    m=lib*li_m;
  }
  void libra_pie(float lib){
    pie=lib*li_pi;
```

```
  }
  void libra_pul(float lib){
    pul=lib*li_pul;
  }
  float lib_met(float m){
    libra_met(m);
    return ret_lib_met();
  }
  float lib_pul(float m){
    libra_pul(m);
    return ret_lib_pul();
  }
  float lib_pie(float m){
    libra_pie(m);
    return ret_lib_pie();
  }
  float ret_lib_pul(){
    return pul;
  }
  float ret_lib_met(){
    return m;
  }
  float ret_lib_pie(){
    return pie;
  }
};
void todo(Metros b,Pulgadas c,Pies d,Libras e);
//////////////////////////
//Funcion Principal
//////////////////////////
int main(){
  Conversion a;
  Metros b;
```

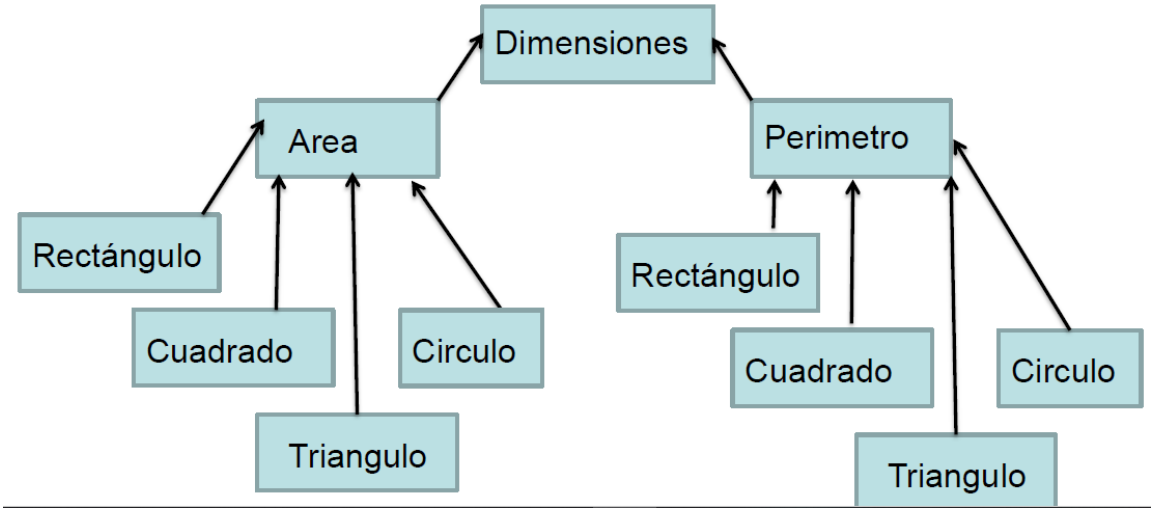
```
  Pulgadas c;
  Pies d;
  Libras e;
  todo(b,c,d,e);

}
void todo(Metros b,Pulgadas c,Pies d,Libras e){
  char a;
  float m,lib,pi,pul;
  char h;
  do{
    cout<<"Dime que quieres convertir"<<endl;
    cout<<"Metros/m, Pies/t, Pulgadas/p, Libras/l"<<endl;
    cin>>a;
    if(a=='m'){
      cout<<"Dime cuantos metros hay:"<<endl;
      cin>>m;
      cout<<"Pies: "<<b.met_pie(m)<<"ft"<<endl;
      cout<<"Pulgadas: "<<b.met_pul(m)<<"in"<<endl;
      cout<<"Libras: "<<b.met_lib(m)<<"lb"<<endl;
    }
    else if(a=='t'){
      cout<<"Dime cuantos pies hay:"<<endl;
      cin>>pi;
      cout<<"Metros: "<<d.pies_met(pi)<<"m"<<endl;
      cout<<"Pulgadas: "<<d.pies_pul(pi)<<"in"<<endl;
      cout<<"Libras: "<<d.pies_lib(pi)<<"lb"<<endl;
    }
    else if(a=='p'){
      cout<<"Dime cuantos pulgadas hay:"<<endl;
      cin>>pul;
      cout<<"Metros: "<<c.pulg_met(pul)<<"m"<<endl;
      cout<<"Pies: "<<c.pulg_pie(pul)<<"ft"<<endl;
```

```
      cout<<"Libras: "<<c.pulg_lib(pul)<<"lb"<<endl;
    }
    else {
      cout<<"Dime cuantos libras hay:"<<endl;
      cin>>lib;
      cout<<"Metros: "<<e.lib_met(lib)<<"m"<<endl;
      cout<<"Pulgadas: "<<e.lib_pul(lib)<<"in"<<endl;
      cout<<"Pies: "<<e.lib_pie(lib)<<"ft"<<endl;
    }
    cout<<"Quieres hacer otras conversion s/n"<<endl;
    cin>>h;
  }while(h!='n');
}
```


problema3
Dime que quieres convertir
Metros/m, Pies/t, Pulgadas/p, Libras/l
m
Dime cuantos metros hay:
10
Pies: 32.8ft
Pulgadas: 393.7in
Libras: 7.376lb
Quieres hacer otras conversion s/n
s
Dime que quieres convertir
Metros/m, Pies/t, Pulgadas/p, Libras/l
t
Dime cuantos pies hay:
10
Metros: 3.048m
Pulgadas: 120in
Libras: 0.83333lb
Quieres hacer otras conversion s/n
s
Dime que quieres convertir
Metros/m, Pies/t, Pulgadas/p, Libras/l
p
Dime cuantos pulgadas hay:
10
Metros: 0.254m
Pies: 0.833333ft
Libras: 20.36lb
Quieres hacer otras conversion s/n
s
Dime que quieres convertir
Metros/m, Pies/t, Pulgadas/p, Libras/l
l
Dime cuantos libras hay:
10
Metros: 0.1152m
Pulgadas: 20.36in
Pies: 0.83333ft
Quieres hacer otras conversion s/n
n
Presione una tecla para continuar . . .

Problema 5:

Elabore un algoritmo que, haciendo uso de herencias, calcule el área y perímetro de las formas indicadas. Ver el siguiente esquema:



Syntaxis:

```
//////////////////////////
//Librerias
//////////////////////////
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <string.h>
using namespace std;
//////////////////////////
//Definiciones
//////////////////////////
#define tam_max 10
//////////////////////////
//Clases
//////////////////////////
class Dimensiones{
public:
```

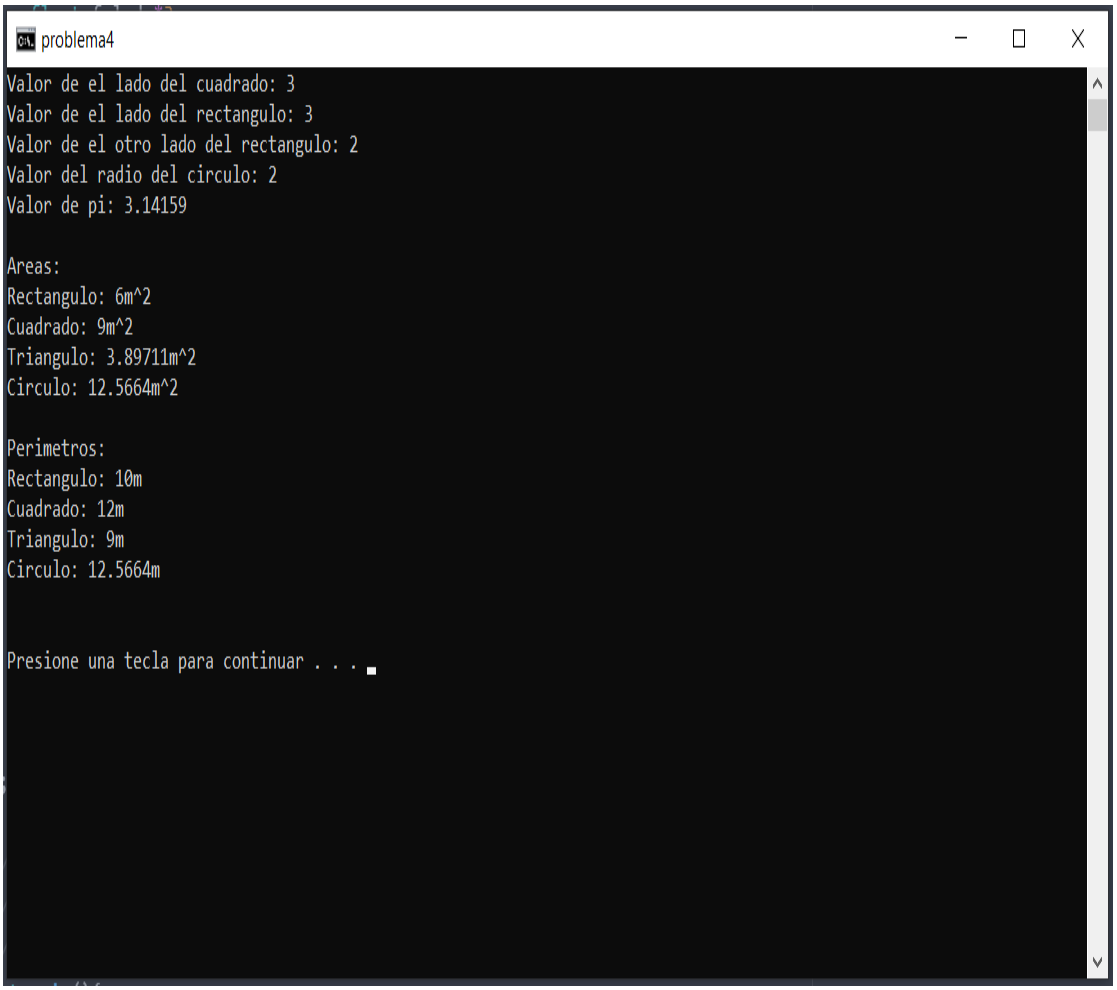
```
  float lado=3;
  float lado2=2;
  float radio=2;
  float altura=5;
  float pi=3.141592654;
  float area;
  float perimetro;
  float res_area(){
    return area;
  }
  float res_peri(){
    return perimetro;
  }
  float valor_s(){
    float s=0;
    s=(lado+lado+lado)/2;
    return s;
  }
  void todo_parametros(){
    cout<<"Valor de el lado del cuadrado: "<<lado<<endl;
    cout<<"Valor de el lado del rectangulo: "<<lado<<endl;
    cout<<"Valor de el otro lado del rectangulo: "<<lado2<<endl;
    cout<<"Valor del radio del circulo: "<<radio<<endl;
    cout<<"Valor de pi: "<<pi<<endl<<endl;
  }
};
class Area:public Dimensiones{
  float rec;
  float cuad;
  float tria;
  float circ;
public:
  void rectangulo(){
```

```
    rec=lado*lado2;
  }
void cuadrado(){
    cuad=lado*lado;
  }
void triangulo(){
    float s=valor_s();
    tria=sqrt(s*(s-lado)*(s-lado)*(s-lado));
  }
void circulo(){
    circ=pi*radio*radio;
  }
  float rectan(){
    rectangulo();
    return ret_rectangulo();
  }
  float cuadra(){
    cuadrado();
    return ret_cuadrado();
  }
  float triang(){
    triangulo();
    return ret_triangulo();
  }
  float circu(){
    circulo();
    return ret_circulo();
  }
  float ret_rectangulo(){
    return rec;
  }
  float ret_cuadrado(){
    return cuad;
```

```
  }
  float ret_triangulo(){
    return tria;
  }
  float ret_circulo(){
    return circ;
  }
  void todo_area(){
    cout<<"Areas: "<<endl;
    area=rectan();
    cout<<"Rectangulo: "<<res_area()<<"m^2"<<endl;
    area=cuadra();
    cout<<"Cuadrado: "<<res_area()<<"m^2"<<endl;
    area=triang();
    cout<<"Triangulo: "<<res_area()<<"m^2"<<endl;
    area=circu();
    cout<<"Circulo: "<<res_area()<<"m^2"<<endl<<endl;
  }
};
class Perimetro:public Dimensiones{
  float rec;
  float cuad;
  float tria;
  float circ;
public:
  void rectangulo(){
    rec=lado*2+lado2*2;
  }
  void cuadrado(){
    cuad=lado*4;
  }
  void triangulo(){
    tria=lado*3;
```

```
}
void circulo(){
  circ=pi*radio*2;
}
float rectan(){
  rectangulo();
  return ret_rectangulo();
}
float cuadra(){
  cuadrado();
  return ret_cuadrado();
}
float triang(){
  triangulo();
  return ret_triangulo();
}
float circu(){
  circulo();
  return ret_circulo();
}
float ret_rectangulo(){
  return rec;
}
float ret_cuadrado(){
  return cuad;
}
float ret_triangulo(){
  return tria;
}
float ret_circulo(){
  return circ;
}
void todo_peri(){
```

```
    cout<<"Perimetros: "<<endl;
    perimetro=rectan();
    cout<<"Rectangulo: "<<res_peri()<<"m"<<endl;
    perimetro=cuadra();
    cout<<"Cuadrado: "<<res_peri()<<"m"<<endl;
    perimetro=triang();
    cout<<"Triangulo: "<<res_peri()<<"m"<<endl;
    perimetro=circu();
    cout<<"Circulo: "<<res_peri()<<"m"<<endl<<endl;
  }
};
//////////////////////////
//Funcion Principal
//////////////////////////
int main(){
  Dimensiones a;
  Area b;
  Perimetro c;
  a.todo_parametros();
  b.todo_area();
  c.todo_peri();
}
```


problema4
Valor de el lado del cuadrado: 3
Valor de el lado del rectangulo: 3
Valor de el otro lado del rectangulo: 2
Valor del radio del circulo: 2
Valor de pi: 3.14159
Areas:
Rectangulo: 6m^2
Cuadrado: 9m^2
Triangulo: 3.89711m^2
Circulo: 12.5664m^2
Perimetros:
Rectangulo: 10m
Cuadrado: 12m
Triangulo: 9m
Circulo: 12.5664m
Presione una tecla para continuar . . .

Conclusiones:

Me pareció una mejora considerable en la elaboración del código, ya que si se desea hacer una clase con los mismos atributos y métodos de otra solo se tiene que heredar el contenido de la clase base.

Así se pude llevar a cabo una reutilización de código y por lo mismo disminuir el número de líneas y trabajo a desempeñar.

Bibliografía:

Concepto de Herencia - Herencia en C++ (Práctica 3). (s. f.). Recuperado 20-11-12, de https://www.codingame.com/playgrounds/50747/herencia-en-c-practica-3/concepto-de-herencia