



Modularización de programas: tipos de módulos (función y procedimiento) y su aplicación en los programas



Propósito

Teoría

Reconoce los tipos de módulos (función y procedimiento) y su aplicación en los programas en el lenguaje de programación



RECUERDA: Uso de Módulos

Observa:

El módulo herencia tiene que calcular de uno hasta dos valores según el if-else (línea 28 y línea 34)

Por lo que se recomienda que sea procedimiento, y no función (ya que solo devuelve un valor)

```
1 #include <iostream>
2 using namespace std;
3
4 void herencia()
5 {
6     float her, may;
7     int cant;
8
9     do{
10         cout << "Ingresar importe de herencia \n";
11         cin >> her;
12
13         if (her <=0)
14             cout << "ERROR.La herencia debe ser mayor a 0";
15     } while (her <=0);
16
17     do{
18         cout << "Ingresar cantidad de hijos \n";
19         cin >> cant;
20
21         if (cant <= 0)
22             cout << "ERROR.La cantidad debe ser mayor a 0";
23     } while (cant <= 0);
24
25     if (cant < 4)
26     {
27         her = her / cant;
28         cout << "La herencia a cada hijo es" << her << endl;
29     }
30     else
31     {
32         may = her / 2;
33         her = may / (cant - 1);
34         cout << "\nLa herencia del hermano mayor es: " << may;
35         cout << "\nLa herencia de los demas hijos es: " << her;
36     }
37 }
38
39 int main()
40 {
41     herencia();
42     return 0;
43 }
```

Observa:

Se implementa un módulo para leer cada tipo de dato, y se invoca en otro módulo, por ejemplo: void herencia()

```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7     cin >> dato;
8     return dato;
9 }
10
11 int leedatoe()
12 {
13     int dato;
14     cin >> dato;
15     return dato;
16 }
17
18 void herencia()
19 {
20     float her, may;
21     int cant;
22
23     do{
24         cout << "Ingresar importe de herencia:\t";
25         her = leedatof();
26
27         if (her <=0)
28             cout << "ERROR.La herencia debe ser mayor a 0";
29     } while (her <=0);
30
31     do{
32         cout << "Ingresar cantidad de hijos:\t";
33         cant = leedatoe();
34
35         if (cant <= 0)
36             cout << "ERROR.La cantidad debe ser mayor a 0";
37     } while (cant <= 0);
38
39     if (cant < 4)
40     {
41         her = her / cant;
42         cout << "La herencia a cada hijo es:\t" << her << endl;
43     }
44     else
45     {
46         may = her / 2;
47         her = may / (cant - 1);
48         cout << "\nLa herencia del hermano mayor es: " << may;
49         cout << "\nLa herencia de los demas hijos es: " << her;
50     }
51 }
52
53 int main()
54 {
55     herencia();
56     return 0;
57 }
```



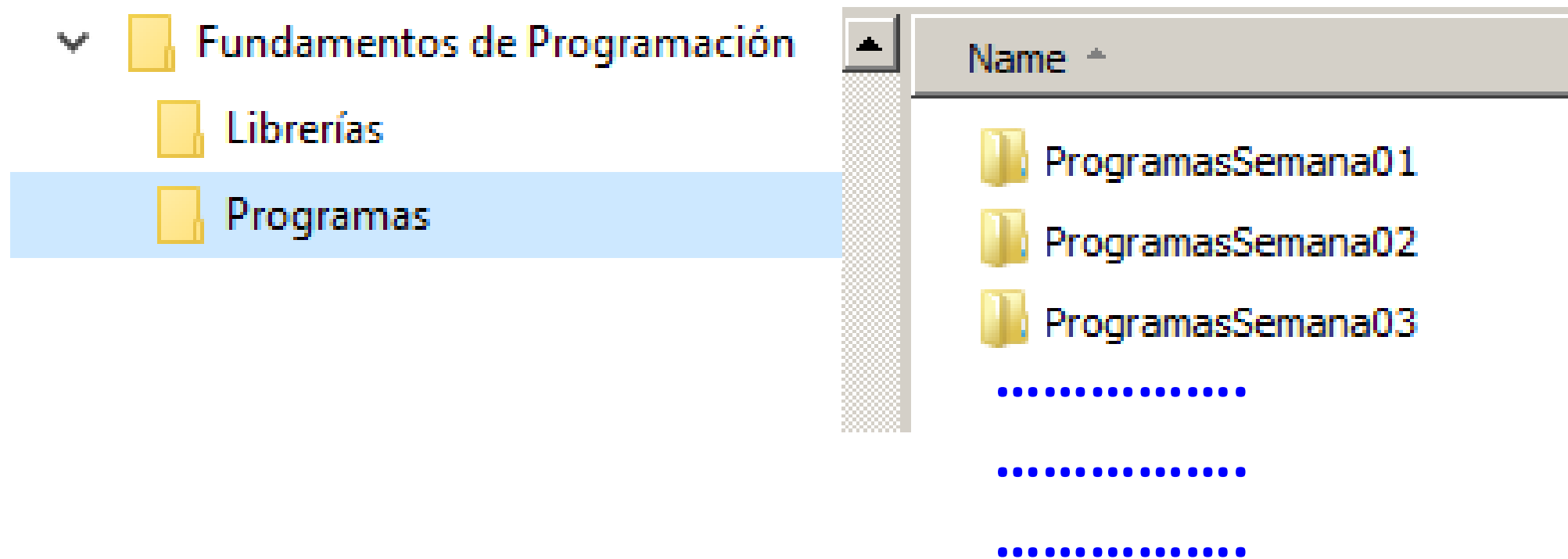
RECUERDA:

Uso de Variables Globales

```
1  #include <iostream>
2  using namespace std;
3
4  int creditos, contEstud, cont = 0, cantFem=0, cantMasc=0, acumCreditos=0;
5  int contFemPrimer=0, contFemSegund=0, contFemTercer=0, contFemCuart=0, contFemQuint=0;
6  int contMascPrimer=0, contMascSegund=0, contMascTercer=0, contMascCuart=0, contMascQuint=0;
7  int acumFemPrimer=0, acumFemSegund=0, acumFemTercer=0, acumFemCuart=0, acumFemQuint=0;
8  int acumMascPrimer=0, acumMascSegund=0, acumMascTercer=0, acumMascCuart=0, acumMascQuint=0;
9  char rpt, genero;
10 string mensaje;
11
12 int leedatoc()
13 {
14     int dato;
15     cin>>dato;
16     return dato;
17 }
18
19 char leedatoc()
20 {
21     char dato;
22     cin>>dato;
23     return dato;
24 }
25
26 void Mostrar()
27 {
28     cout<<"\nLa cantidad de estudiantes es:\t" << contEstud <<"\n";
29     cout<<"\nEl total acumulado de creditos es:\t" << acumCreditos <<"\n";
30     cout<<"\nLa cantidad de estudiantes femenino en primero es:\t" << contFemPrimer <<"\n";
31     cout<<"\nLa cantidad de estudiantes femenino en segundo es:\t" << contFemSegund <<"\n";
32     cout<<"\nLa cantidad de estudiantes femenino en tercero es:\t" << contFemTercer <<"\n";
33     cout<<"\nLa cantidad de estudiantes femenino en cuarto es:\t" << contFemCuart <<"\n";
34     cout<<"\nLa cantidad de estudiantes femenino en quinto es:\t" << contFemQuint <<"\n";
35     cout<<"\nLa cantidad de estudiantes masculino en primero es:\t" << contMascPrimer <<"\n";
36     cout<<"\nLa cantidad de estudiantes masculino en segundo es:\t" << contMascSegund <<"\n";
37     cout<<"\nLa cantidad de estudiantes masculino en tercero es:\t" << contMascTercer <<"\n";
38     cout<<"\nLa cantidad de estudiantes masculino en cuarto es:\t" << contMascCuart <<"\n";
39     cout<<"\nLa cantidad de estudiantes masculino en quinto es:\t" << contMascQuint <<"\n";
40     cout<<"\nEl acumulado de creditos femenino en primero es:\t" << acumFemPrimer <<"\n";
41     cout<<"\nEl acumulado de creditos femenino en segundo es:\t" << acumFemSegund <<"\n";
42     cout<<"\nEl acumulado de creditos femenino en tercero es:\t" << acumFemTercer <<"\n";
43     cout<<"\nEl acumulado de creditos femenino en cuarto es:\t" << acumFemCuart <<"\n";
44     cout<<"\nEl acumulado de creditos femenino en quinto es:\t" << acumFemQuint <<"\n";
45     cout<<"\nEl acumulado de creditos masculino en primero es:\t" << acumMascPrimer <<"\n";
46     cout<<"\nEl acumulado de creditos masculino en segundo es:\t" << acumMascSegund <<"\n";
47     cout<<"\nEl acumulado de creditos masculino en tercero es:\t" << acumMascTercer <<"\n";
48     cout<<"\nEl acumulado de creditos masculino en cuarto es:\t" << acumMascCuart <<"\n";
49     cout<<"\nEl acumulado de creditos masculino en quinto es:\t" << acumMascQuint <<"\n";
50 }
```

Variables Globales

Recomendación para crear los programas:



Ahora, elaboremos los programas ejemplos..!



Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto1_HerenciaModulo

Nombre Programa: Programa1_HerenciaModulo

```
1  #include <iostream>
2  using namespace std;
3
4  float leedatof()
5  {
6      float dato;
7
8      cin>>dato;
9
10     return dato;
11 }
12
13 int leedatof()
14 {
15     int dato;
16
17     cin>>dato;
18
19     return dato;
20 }
21
22 void herencia()
23 {
24     float her, may;
25     int cant;
26
27     do{
28         cout << "Ingresar importe de herencia:\t";
29         her = leedatof();
30
31         if (her <= 0)
32             cout << "ERROR.La herencia debe ser mayor a 0";
33     } while (her <= 0);
34
35     do{
36         cout << "Ingresar cantidad de hijos:\t";
37         cant = leedatof();
38
39         if (cant <= 0)
40             cout << "ERROR.La cantidad debe ser mayor a 0";
41
42     } while (cant <= 0);
43
44     if (cant < 4)
45     {
46         her = her / cant;
47         cout << "La herencia a cada hijo es:\t" << her << endl;
48     }
49     else
50     {
51         may = her / 2;
52         her = may / (cant - 1);
53         cout << "\nLa herencia del hermano mayor es: " << may;
54         cout << "\nLa herencia de los demas hijos es: " << her;
55     }
56 }
57
58
59 int main()
60 {
61     herencia();
62
63     return 0;
64 }
```



Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto2_IntercambioNumerosModulo

Nombre Programa: Programa2_IntercambioNumerosModulo

```
1  #include<iostream>
2  using namespace std;
3
4  int leedatoe()
5  {
6      int dato;
7
8      cin>>dato;
9
10     return dato;
11 }
12
13 void Intercambio()
14 {
15     int num1, num2, temp;
16
17     cout<<"Ingrese valor de num1: \t";
18     num1 = leedatoe();
19
20     cout<<"Ingrese valor de num2: \t";
21     num2 = leedatoe();
22
23     temp = num1;
24     num1 = num2;
25     num2 = temp;
26
27     cout<<"\n";
28     cout<<"El nuevo valor num1 es:\t" << num1 <<endl;
29     cout<<"El nuevo valor num2 es:\t" << num2 <<endl;
30
31 }
32
33 int main()
34 {
35     Intercambio();
36
37     return 0;
38 }
```



Carpeta: Programas Semana 10

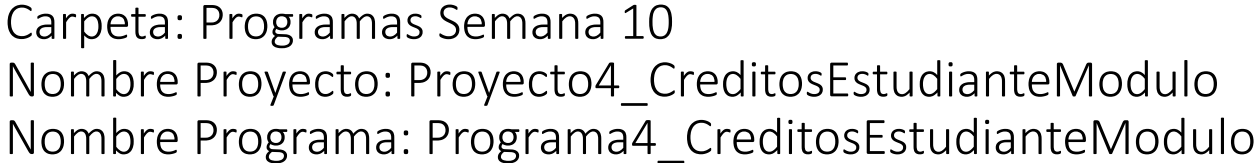
Nombre Proyecto: Proyecto3_MayorMenorNumerosModulo

Nombre Programa: Programa3_MayorMenorNumerosModulo

```
1  #include <iostream>
2  using namespace std;
3
4  int cont = 0, may, men, dato; // declaracion global
5  char rpt;
6
7  int leedatoe()
8  {
9      int dato;
10     cin>>dato;
11     return dato;
12 }
13
14 char leedatoc()
15 {
16     char dato;
17     cin>>dato;
18     return dato;
19 }
20
21 void Mostrar()
22 {
23     cout<<"\nEl mayor es:\t" << may <<"\n";
24     cout<<"\nEl menor es:\t" << men <<"\n";
25 }
26
27 void MayorMenor()
28 {
29     do{
30         cout<<"\nIngrese valor:\t";
31         dato = leedatoe();
32
33         cont = cont + 1; // cont++;
34
35         if(cont == 1)
36         {
37             may = dato;
38             men = dato;
39         }
40         else
41         {
42             if(dato > may)
43                 may = dato;
44             else
45                 if(dato < men)
46                     men = dato;
47         }
48         do{
49             cout<<"\nDesea ingresar otros numero? (Ss - Nn):\t";
50             rpt = leedatoc();
51
52             if(rpta != 'S' && rpta != 's' && rpta != 'N' && rpta != 'n')
53                 cout<<"\nERROR. Vuelva a ingresar S o s o N o n.\n";
54
55             }while(rpta != 'S' && rpta != 's' && rpta != 'N' && rpta != 'n' );
56         }while(rpta == 'S' || rpta == 's');
57     }
58     Mostrar();
59 }
60
61 int main()
62 {
63     MayorMenor();
64 }
```

ucontinental.edu.pe





Créditos acumulados	Año académico
Menos de 32	Primero
De 32 a 63	Segundo
De 64 a 95	Tercero
De 96 a 127	Cuarto
De 128 a más	Quinto



```

107         case 'm':{
108
109
110             if (creditos < 32)
111             {
112                 mensaje = "Primero";
113                 contMascPrimer++;
114                 acumMascPrimer = acumMascPrimer + creditos;
115             }
116             else
117             {
118                 if (creditos >=32 && creditos <= 63)
119                 {
120                     mensaje = "Segundo";
121                     contMascSegund++;
122                     acumMascSegund = acumMascSegund + creditos;
123                 }
124                 else
125                 {
126                     if (creditos >=64 && creditos <= 95)
127                     {
128                         mensaje = "Tercero";
129                         contMascTercer++;
130                         acumMascTercer = acumMascTercer + creditos;
131                     }
132                     else
133                     {
134                         if (creditos >=96 && creditos <= 127)
135                         {
136                             mensaje = "Cuarto";
137                             contMascCuart++;
138                             acumMascCuart = acumMascCuart + creditos;
139                         }
140                         else
141                         {
142                             if (creditos >=128 )
143                             {
144                                 mensaje = "Quinto";
145                                 contMascQuint++;
146                                 acumMascQuint = acumMascQuint + creditos;
147                             }
148                         }
149                     }
150                 }
151             }
152             break;
153         }
154
155         cout<<"\nEl estudiante es de:\t" << mensaje <<"\n";
156
157         contEstud++; // contador general
158         acumCreditos = acumCreditos + creditos; // acumulador general
159
160         do{
161             cout<<"\nDesea ingresar otro estudiante? (Ss - Nn):\t";
162             rpta = leedatoc();
163             if(rpta != 'S' && rpta != 's' && rpta != 'N' && rpta != 'n')
164                 cout<<"\nERROR. Vuelva a ingresar S o s o N o n.\n";
165             while(rpta != 'S' && rpta != 's' && rpta != 'N' && rpta != 'n' );
166
167         }while(rpta == 'S' || rpta == 's');
168
169         Mostrar();
170     }
171
172     int main()
173     {
174         ProcesaCreditos();
175
176         return 0;
177     }
178 }

```





Conclusiones

Modularización
de programas:
tipos de
módulos y su
aplicación en
los programas

Algunos módulos
deben ser
función, como los
lee datos.

El módulo es
procedimiento si
debe calcular y
mostrar mas
de un valor.

El módulo
principal invoca
a un módulo
que realiza el
algoritmo de
solución.

Programa

```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7     cin >> dato;
8     return dato;
9 }
10
11 int leedatoe()
12 {
13     int dato;
14     cin >> dato;
15     return dato;
16 }
17
18 void herencia()
19 {
20     float her, may;
21     int cant;
22
23     do{
24         cout << "Ingresar importe de herencia:\t";
25         her = leedatof();
26         if (her <= 0)
27             cout << "ERROR.La herencia debe ser mayor a 0";
28     } while (her <= 0);
29
30     do{
31         cout << "Ingresar cantidad de hijos:\t";
32         cant = leedatoe();
33         if (cant <= 0)
34             cout << "ERROR.La cantidad debe ser mayor a 0";
35     } while (cant <= 0);
36
37     if (cant < 4)
38     {
39         her = her / cant;
40         cout << "La herencia a cada hijo es:\t" << her << endl;
41     }
42     else
43     {
44         may = her / 2;
45         her = may / (cant - 1);
46         cout << "\nLa herencia del hermano mayor es: " << may;
47         cout << "\nLa herencia de los demas hijos es: " << her;
48     }
49 }
50
51 int main()
52 {
53     herencia();
54     return 0;
55 }
```

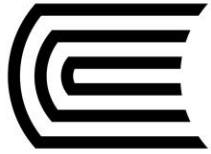




Propósito

Práctica

Crea programas con tipos de módulos (función y procedimiento), en el lenguaje de programación C/ C++.



Desarrollamos la Guía Práctica 10

Guía práctica N°10

Fundamentos de Programación

Tercera Unidad: Módulos para la programación: función y procedimiento

Sección :	Apellidos :
Docente :	Nombres :
	Fecha : / / Duración: 100 min
	Tipo de práctica: Individual (x) Grupal ()

Instrucciones: Lea detenidamente cada enunciado y desarrolle lo solicitado.
Utilizar el Dev C++ para el desarrollo de los siguientes programas

1. Elaborar un programa, para una empresa que otorga aumento al sueldo base de su trabajador (ver tabla). Crear un módulo `ReporteFinal()`, para mostrar la cantidad total de trabajadores, la cantidad de trabajadores que recibieron aumento de 10%, la cantidad de trabajadores que recibieron aumento de 15%, la cantidad de trabajadores que recibieron aumento de 20%, y la cantidad de trabajadores que recibieron aumento de 25%, además del acumulado de sueldos totales. Puedes usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso. Referencia de solución:

Carpeta: Programas Semana 09
Nombre Proyecto: Proyecto7_AumentoSueldoModulo
Nombre Programa: Programa7_AumentoSueldoModulo

Sueldo (\$.)	Aumento (%)
5001 - más	10
3501 - 5000	15
2001 - 3500	20
hasta 2000	25

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto5_AumentoSueldoModuloV2

Nombre Programa: Programa5_AumentoSueldoModuloV2

2. Elaborar un programa, para ingresar género y estado civil de una persona, así como su respectiva nota (0 a 20) y crear el módulo `ReporteFinalEstudiante()` para mostrar la siguiente información. La cantidad de hombres aprobados.

La cantidad de mujeres aprobadas.
La cantidad de hombres casados desaprobados.
La cantidad de mujeres solteras aprobadas.
La sumatoria de todas las notas ingresadas.
El promedio de notas ingresadas.

Referencia de Solución:

Carpeta: Programas Semana 07
Nombre Proyecto: Proyecto4_NotasContaAcumMas
Nombre Programa: Programa4_NotasContaAcumMas

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto6_ContaAcumulaModulo

Nombre Programa: Programa6_ContaAcumulaModulo

3. Elaborar un programa para gestionar las facturas, para una empresa que se dedica a la venta de desinfectantes. Para cada factura se ingresa: el código del artículo, la cantidad vendida en litros y el precio por litro. Mostrar por cada factura: el monto total a facturar (monto de pago), y además crear un módulo para mostrar la cantidad total de facturas ingresadas, la cantidad de facturas con monto total a facturar de más de 600, el acumulado de monto total a facturar, y el acumulado de litros vendidos. Puedes usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto7_DesinfectantesModulo

Nombre Programa: Programa7_DesinfectantesModulo

4. Elaborar un programa que que determine el IMC (índice de masa corporal) de una persona al ingresar su peso en kg y su talla en metros ($IMC = \text{Peso} / \text{Talla}^2$), y por cada persona mostrar su estado según la siguiente tabla:

Resultado IMC	Estado
Menos de 18.50	Infra Peso
De 18.50 a 24.99	Peso Normal
De 25 a 29.99	Sobre Peso
De 30 a 34.99	Obesidad leve
De 35 a 39.99	Obesidad media
De 40 a más	Obesidad mórbida

Para la validación de datos, se considera:
Peso mínimo: 38 kg - Peso máximo: 260 kg.
Talla mínima 0.90 m - Talla máxima 2.20 m.

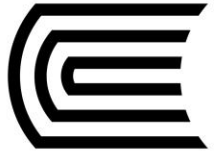
Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto8_MasaCorporalModulo

Nombre Programa: Programa8_MasaCorporalModulo

Además, crear un módulo para mostrar la cantidad de personas ingresadas, la cantidad de personas con estado "Peso Normal", la cantidad de personas con "Obesidad Media" y la cantidad de personas con "Obesidad Mórbida".

Puedes usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso.



Conclusiones

Modularización
de programas:
tipos de
módulos y su
aplicación en
los programas

Algunos módulos
deben ser
función, como los
lee datos.

El módulo es
procedimiento
si debe calcular
y mostrar mas
de un valor.

El módulo
principal invoca
a un módulo
que realiza el
algoritmo de
solución.

Programa

```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7     cin>>dato;
8     return dato;
9 }
10
11 int leedatoe()
12 {
13     int dato;
14     cin>>dato;
15     return dato;
16 }
17
18 void herencia()
19 {
20     float her, may;
21     int cant;
22
23     do{
24         cout << "Ingresar importe de herencia:\t";
25         her = leedatof();
26         if (her <= 0)
27             cout << "ERROR.La herencia debe ser mayor a 0";
28     } while (her <= 0);
29
30     do{
31         cout << "Ingresar cantidad de hijos:\t";
32         cant = leedatof();
33         if (cant <= 0)
34             cout << "ERROR.La cantidad debe ser mayor a 0";
35     } while (cant <= 0);
36
37     if (cant < 4)
38     {
39         her = her / cant;
40         cout << "La herencia a cada hijo es:\t" << her << endl;
41     }
42     else
43     {
44         may = her / 2;
45         her = may / (cant - 1);
46         cout << "\nLa herencia del hermano mayor es: " << may;
47         cout << "\nLa herencia de los demas hijos es: " << her;
48     }
49 }
50
51 int main()
52 {
53     herencia();
54     return 0;
55 }
```





ucontinental.edu.pe