

RESOLVER LOS EJERCICIOS DESDE LA SECCION 4.11 HASTA LA SECCION 4.36

Ejercicios

4.11 Identifique y corrija los errores en cada uno de los siguientes fragmentos de código:

a)

```
int main()
{
    if ( edad >= 65 );
    cout << "Edad es mayor o igual que 65" << endl;
    else
    cout << "Edad es menor que 65 << endl";
    return 0;
}
```

CÓDIGO CORREGIDO:

```
int main()
{
    int edad;
    if ( edad >= 65 )
    cout << "Edad es mayor o igual que 65" << endl;
    else
    cout << "Edad es menor que 65 << endl";
    return 0;
}
```

b)

```
if ( edad >= 65 );  
  
cout << "Edad es mayor o igual que 65 << endl";  
  
else;  
  
cout << "Edad es menor que 65 << endl";
```

CÓDIGO CORREGIDO:

```
int edad;  
  
if ( edad >= 65 )  
  
cout << "Edad es mayor o igual que 65" << endl;  
  
else  
  
cout << "Edad es menor que 65 << endl";
```

c)

```
int x = 1, total;  
  
while ( x <= 10 )  
  
    {  
  
        total += x;  
  
        ++x;  
  
    }
```

CÓDIGO CORREGIDO:

```
int x = 1, total;  
  
while ( x <= 10 )  
  
    {  
  
        total+= x;  
  
        cout<<++x<<endl;  
  
    }
```

d)

```
while ( x <= 100 )
```

```
total += x;
```

```
++x;
```

CÓDIGO CORREGIDO:

```
int x = 1, total;
```

```
while ( x <= 100 )
```

```
{
```

```
    total += x;
```

```
    cout<<++x<<endl;
```

```
}
```

e)

```
while ( y > 0 )
```

```
{
```

```
    cout << y << endl;
```

```
    ++y;
```

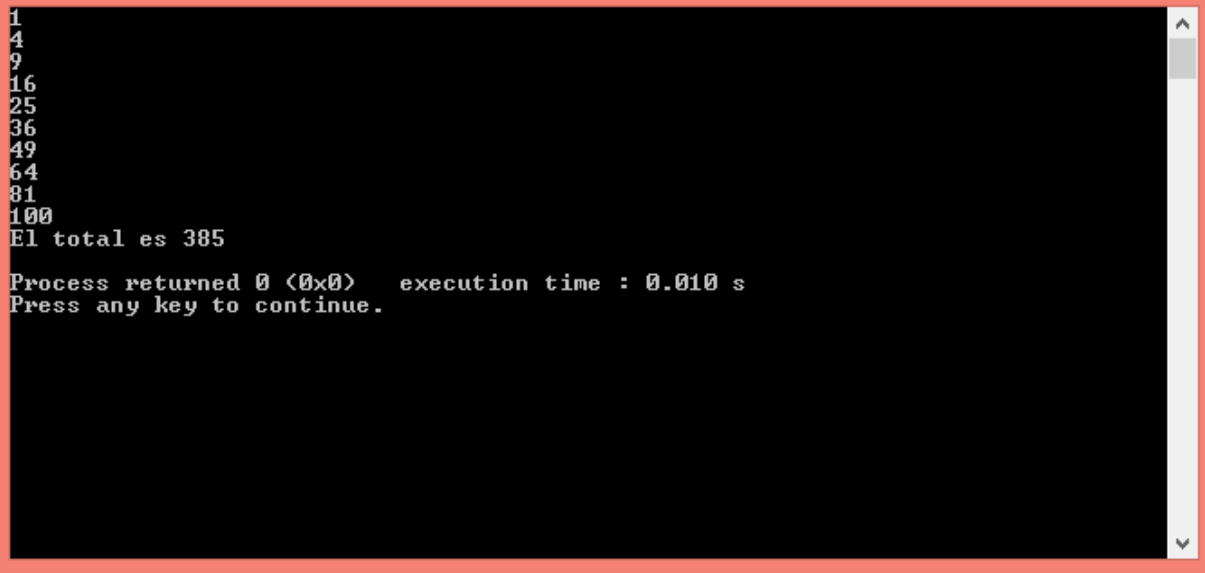
```
}
```

CORRECCIÓN: Se debe primeramente declarar y e inicializándola en un valor, y establecer al while un "tope" máximo hasta el cual contar, caso contrario se quedara incrementando indefinidamente.

4.12 ¿Qué es lo que imprime el siguiente programa?

```
1 // Ejercicio 4.12: ej04_12.cpp
2 // ¿Qué imprime este programa?
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     int y; // declara y
10    int x = 1; // inicializa x
11    int total = 0; // inicializa el total
12
13    while ( x <= 10 ) // itera 10 veces
14    {
15        y = x * x; // realiza el cálculo
16        cout << y << endl; // imprime el resultado
17        total += y; // suma y al total
18        x++; // incrementa el contador x
19    } // fin de while
20
21    cout << "El total es " << total << endl; // muestra el resultado
22    return 0; // indica que terminó correctamente
23 } // fin de main
```

Respuesta: Imprime una serie de números desde el 1 hasta el 100, empezando en 1 y sumando el número primo siguiente correspondiente que pertenezcan al intervalo 1 a 100.



```
1
4
9
16
25
36
49
64
81
100
El total es 385

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

Para los ejercicios 4.13 a 4.16, realice cada uno de los siguientes pasos:

- a) Lea el enunciado del problema.
- b) Formule el algoritmo utilizando pseudocódigo y el proceso de mejoramiento de arriba a abajo, paso a paso.
- c) Escriba un programa en C++.
- d) Pruebe, depure y ejecute el programa en C++.

Los conductores se preocupan acerca del kilometraje de sus automóviles. Un conductor ha llevado el registro de varios reabastecimientos de gasolina, registrando los kilómetros conducidos y los litros usados en cada reabastecimiento. Desarrolle un programa en C++ que utilice una instrucción while para recibir como entrada los kilómetros conducidos y los litros usados (ambos como enteros) por cada reabastecimiento. El programa debe calcular y mostrar los kilómetros por litro obtenidos en cada reabastecimiento, y debe imprimir el total de kilómetros por litro obtenidos en todos los reabastecimientos hasta este punto.

b) ELABORAR EL PSEUDOCÓDIGO

Hacer

```
{
    Escribir "Ingrese los litros utilizados: ";
    Leer litros;
    Si litros diferente de -1
    {
        Escribir "Escriba los kilómetros usados: ";
        Leer "distancia"
        resultado=distancia/litros;
        total litros = total litros +litros;
        total kilómetros = total kilómetros+distancia ;
        promedio total= total kilómetros/total litros;
        Escribir "KPL en este reabastecimiento: " << resultado;
    }
}
Mientras litros sea diferente de -1;
Escribir "Total KPL: " << promedio;
Fin
```

c) Escribir un programa en C++

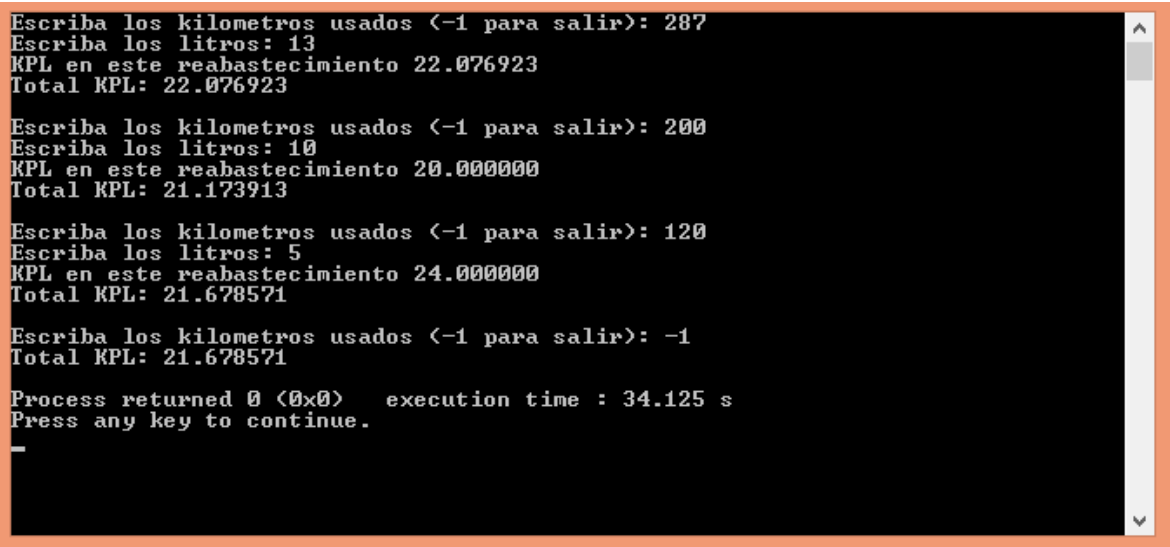
```
#include <iostream>
using namespace std;
#include <iomanip>
using std::setprecision;
int main()
{
    double distancia; // declara distancia
    double litros; // declara litros
    double resultado; // declara resultado
    double totalkms = 0; // declara kilometros totales
    double totLitros = 0;
    double prom = 0; // inicializa total litros

    do
    {
        cout << "Escriba los kilometros usados (-1 para salir): ";
        cin >> distancia;
        if ( distancia != -1 )
        {
            cout << "Escriba los litros: ";
            cin >> litros;
            resultado = distancia / litros;
            totLitros += litros;
            totalkms += distancia;
            prom = totalkms / totLitros;
            cout << fixed << setprecision ( 6 );
            cout << "KPL en este reabastecimiento " << resultado << endl;
            cout << "Total KPL: " << prom << endl << endl;
        }

    } while ( distancia != -1 );

    cout << "Total KPL: " << prom << endl;
    return 0;
}
```

d) Pruebe, depure y ejecute el programa en C++



```
Escriba los kilometros usados (-1 para salir): 287
Escriba los litros: 13
KPL en este reabastecimiento 22.076923
Total KPL: 22.076923

Escriba los kilometros usados (-1 para salir): 200
Escriba los litros: 10
KPL en este reabastecimiento 20.000000
Total KPL: 21.173913

Escriba los kilometros usados (-1 para salir): 120
Escriba los litros: 5
KPL en este reabastecimiento 24.000000
Total KPL: 21.678571

Escriba los kilometros usados (-1 para salir): -1
Total KPL: 21.678571

Process returned 0 (0x0)   execution time : 34.125 s
Press any key to continue.
```

4.14 Desarrolle una aplicación en C++ que determine si alguno de los clientes de una tienda de departamentos se ha excedido del límite de crédito en una cuenta. Para cada cliente se tienen los siguientes datos:

- a) Número de cuenta (un entero)
- b) Saldo al inicio del mes
- c) Total de todos los artículos cargados por el cliente en el mes
- d) Total de todos los créditos aplicados a la cuenta del cliente en el mes
- e) Límite de crédito permitido.

El programa debe usar una instrucción while para recibir como entrada cada uno de estos datos, debe calcular el nuevo saldo (= saldo inicial + cargos – créditos) y determinar si éste excede el límite de crédito del cliente. Para los clientes cuyo límite de crédito sea excedido, el programa debe mostrar el número de cuenta del cliente, su límite de crédito, el nuevo saldo y el mensaje "Se excedió el límite de su crédito".

b) Formule el algoritmo utilizando pseudocódigo y el proceso de mejoramiento de arriba a abajo, paso a paso.

HACER

```
{
    Escribir "Introduzca el numero de la cuenta: ";
    Leer cuenta;
    Si cuenta diferente de -1
    {
        Escribir "Introduzca el saldo inicial: ";
        Leer saldo inicial;
        Escribir "Introduzca los cargos totales: ";
        Leer cargos;
        Escribir "Introduzca los creditos totales: ";
        Leer creditos;
        Escribir "Introduzca el limite de credito: ";
        Leer limite;
        nuevo saldo = saldo inicial + cargos - creditos;
        SI (nuevo saldo > limite)
        {
            Escribir "El nuevo saldo es: "<<nuevo saldo;
            Escribir "Cuenta: "<<cuenta;
            Escribir "Limite de credito: "<<limite;
            Escribir "Saldo: "<<saldo;
            Escribir "Se excedio el limite de su credito: ";
        }
        FIN SI
        CASO CONTRARIO
        {
            Escribir "El nuevo saldo es: "<<nuevo saldo;
        }
    }
}
MIENTRAS (cuenta sea diferente de -1);
}
```

FIN

c) Escriba un programa en C++.

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

#include <iomanip>
using std::setprecision;

int main ()
{
    int cuenta;
    float saldoi, cargos, creditos, limite, nuevosaldo;
    do
    {
        cout << "Introduzca el numero de cuenta (o -1 para salir): ";
        cin >> cuenta;

        if ( cuenta != -1 )
        {
            cout << "Introduzca el saldo inicial: ";
            cin >> saldoi;
            cout << "Introduzca los cargos totales: ";
            cin >> cargos;
            cout << "Introduzca los creditos totales: ";
            cin >> creditos;
            cout << "Introduzca el limite de credito:";
            cin >> limite;
            nuevosaldo = saldoi + cargos - creditos;
            cout << fixed << setprecision ( 2 );
            if (nuevosaldo > limite)
            {
                cout << "\n\nEl nuevo saldo es: "<<nuevosaldo<<endl;
                cout << "Cuenta: "<<cuenta;
                cout << "\nLimite de credito: "<<limite;
                cout << "\nSaldo: "<<nuevosaldo<<endl;
                cout << "Se excedio el limite de su credito."<<endl<<endl<<endl;
            }

            else
            {
                cout <<"El nuevo saldo es: "<<nuevosaldo<<endl<<endl;
            }
        }
    }while(cuenta !=-1);

    return 0;
}
```


d) Pruebe, depure y ejecute el programa en C++.

```
Introduzca el numero de cuenta (o -1 para salir): 100
Introduzca el saldo inicial: 5394.78
Introduzca los cargos totales: 1000.00
Introduzca los creditos totales: 500.00
Introduzca el limite de credito: 5500.00

El nuevo saldo es: 5894.78
Cuenta: 100
Limite de credito: 5500.00
Saldo: 5894.78
Se excedio el limite de su credito.

Introduzca el numero de cuenta (o -1 para salir): 200
Introduzca el saldo inicial: 1000.00
Introduzca los cargos totales: 123.45
Introduzca los creditos totales: 321.00
Introduzca el limite de credito: 1500.00
El nuevo saldo es: 802.45

Introduzca el numero de cuenta (o -1 para salir): 300
Introduzca el saldo inicial: 500.00
Introduzca los cargos totales: 274.73
Introduzca los creditos totales: 100.00
Introduzca el limite de credito: 800.00
El nuevo saldo es: 674.73

Introduzca el numero de cuenta (o -1 para salir): -1
Process returned 0 (0x0)   execution time : 73.902 s
Press any key to continue.
```

4.15 Una gran empresa de químicos paga a sus vendedores mediante comisiones. Los vendedores reciben \$200 por semana, más el 9% de sus ventas brutas durante esa semana. Por ejemplo, un vendedor que vende \$5000 de mercancía en una semana, recibe \$200 más el 9% de \$5000, o un total de \$650. Desarrolle un programa en C++ que utilice una instrucción while para recibir como entrada las ventas brutas de cada vendedor de la semana anterior, y que calcule y muestre los ingresos de ese vendedor. Procese las cifras de un vendedor a la vez.

b) Formule el algoritmo utilizando pseudocódigo y el proceso de mejoramiento de arriba a abajo, paso a paso.

INICIO

sueldo, salario, ventas reales

HACER

```
{
    sueldo=200
    ventas=0
    salario=0
    Escribir "Introduzca las ventas en dolares (o -1 para salir): "
    Leer ventas
    SI (ventas diferente de -1)
    {
        salario=sueldo+(ventas*0.09)
        Escribir "El salario es: "<<salario;
    }
}
```

MIENTRAS (ventas!= -1);

FIN

c) Escriba un programa en C++.

```
#include<iostream>
```

```
using namespace std;
```

```
#include <iomanip>
```

```
using std::setprecision;
```

```
int main()
```

```
{
```

```
    float sueldo, salario, ventas;
```

```
    do
```

```
    {
```

```
        sueldo=200;
```

```
        ventas=0;
```

```
        salario=0;
```

```
        cout<<"Introduzca las ventas en dolares (-1 para salir): ";
```

```
        cin>>ventas;
```

```
        if(ventas!=-1)
```

```
        {
```

```
            salario=sueldo+(ventas*0.09);
```

```
            cout << fixed << setprecision ( 2 );
```

```
            cout<<"El salario es: "<<salario<<endl<<endl;
```

```
        }
```

```
    }
```

```
    while(ventas!=-1);
```

```
    return 0;
```

```
}
```

d) Pruebe, depure y ejecute el programa en C++.

```
Introduzca las ventas en dolares <-1 para salir>: 5000.00
El salario es: 650.00

Introduzca las ventas en dolares <-1 para salir>: 6000.00
El salario es: 740.00

Introduzca las ventas en dolares <-1 para salir>: 7000.00
El salario es: 830.00

Introduzca las ventas en dolares <-1 para salir>: -1
Process returned 0 (0x0)   execution time : 9.848 s
Press any key to continue.
```

4.16 Desarrolle un programa en C++ que utilice una instrucción while para determinar el sueldo bruto para cada uno de varios empleados. La empresa paga la cuota normal en las primeras 40 horas de trabajo de cada empleado, y paga cuota y media en todas las horas trabajadas que excedan de 40. Usted recibe una lista de los empleados de la empresa, el número de horas que trabajó cada empleado la semana pasada y la tarifa por horas de cada empleado. Su programa debe recibir como entrada esta información para cada empleado, debe determinar y mostrar el sueldo bruto de cada empleado.

b) Formule el algoritmo utilizando pseudocódigo y el proceso de mejoramiento de arriba a abajo, paso a paso.

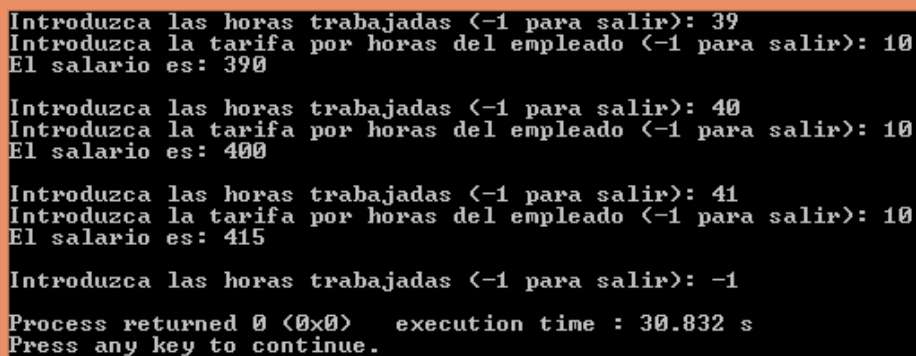
```
INICIO
    HACER
        {
            Escribir "Introduzca las horas trabajadas (-1 para salir): ";
            Leer horas;
            SI (horas diferente de -1)
            {
                SI (horas <= 40)
                {
                    Escribir<<"Introduzca la tarifa por horas del empleado (-1 para salir): ";
                    Leer tarifa;
                    salario = horas*tarifa;
                    Escribir "El salario es: "<<salario;
                }
                SI (horas >40)
                {
                    Escribir "Introduzca la tarifa por horas del empleado (-1 para salir): ";
                    Leer tarifa;
                    extra = (horas*tarifa)+(tarifa*0.5);
                    Escribir "El salario es: "<<extra;
                }
            }
        }
    } MIENTRAS (horas diferente de -1);
FIN
```

c) Escriba un programa en C++.

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

int main()
{
    int horas=0;
    float tarifa=0, salario=0,extra=0;
    do
    {
        cout<<"Introduzca las horas trabajadas (-1 para salir): ";
        cin>>horas;
        if(horas!=-1)
        {
            if (horas<=40)
            {
                cout<<"Introduzca la tarifa por horas del empleado (-1 para salir): ";
                cin>>tarifa;
                salario=horas*tarifa;
                cout<<"El salario es: "<<salario<<endl<<endl;
            }
            if (horas >40)
            {
                cout<<"Introduzca la tarifa por horas del empleado (-1 para salir): ";
                cin>>tarifa;
                extra=(horas*tarifa)+(tarifa*0.5);
                cout<<"El salario es: "<<extra<<endl<<endl;
            }
        }
    } while(horas !=-1);
    return 0;
}
```

d) Pruebe, depure y ejecute el programa en C++.



```
Introduzca las horas trabajadas (-1 para salir): 39
Introduzca la tarifa por horas del empleado (-1 para salir): 10
El salario es: 390

Introduzca las horas trabajadas (-1 para salir): 40
Introduzca la tarifa por horas del empleado (-1 para salir): 10
El salario es: 400

Introduzca las horas trabajadas (-1 para salir): 41
Introduzca la tarifa por horas del empleado (-1 para salir): 10
El salario es: 415

Introduzca las horas trabajadas (-1 para salir): -1
Process returned 0 (0x0)   execution time : 30.832 s
Press any key to continue.
-
```

4.17 El proceso de encontrar el número más grande (es decir, el máximo de un grupo de números) se utiliza frecuentemente en aplicaciones de computadora. Por ejemplo, un programa para determinar el ganador de un concurso de ventas recibe como entrada el número de unidades vendidas por cada vendedor. El vendedor que haya vendido más unidades es el que gana el concurso.

Escriba un programa en pseudocódigo y después una aplicación en C++ que utilice una instrucción *while* para determinar e imprimir el mayor número de una serie de 10 números introducidos por el usuario. Su programa debe utilizar tres variables, como se muestra a continuación:

Contador: Un contador para contar hasta 10 (es decir, para llevar el registro de cuántos números se han introducido, y para determinar cuándo se hayan procesado los 10 números).

Numero: El número actual que se introduce al programa.

Mayor: El número más grande encontrado hasta ahora.

a) Escribir el pseudocódigo del programa

INICIO

MIENTRAS (contador <=10)

{

 Escribir "Por favor introduzca el numero de ventas del vendedor "<<contador++<<": ";

 Leer numero;

 SI (numero > mayor)

 mayor = numero;

}

Escribir "El ganador del concurso es quien logro vender: "<<mayor<<" unidades"

FIN

b) Escribir una aplicación en C++

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int numero,contador = 1, mayor=0;
```

```
    while (contador <=10)
```

```
    {
```

```
        cout<<"\nPor favor introduzca el numero de ventas del vendedor "<<contador++<<": ";
```

```
        cin>>numero;
```

```
        if ( numero > mayor)
```

```
            mayor = numero;
```

```
    }
```

```
    cout<<"\nEl ganador del concurso es quien logro vender: "<<mayor<<" unidades"<<endl<<endl;
```

```
    return 0;
```

```
}
```

c) Pruebe, depure y ejecute el programa en C++.

```
Por favor introduzca el numero de ventas del vendedor 1: 50
Por favor introduzca el numero de ventas del vendedor 2: 655
Por favor introduzca el numero de ventas del vendedor 3: 65
Por favor introduzca el numero de ventas del vendedor 4: 32
Por favor introduzca el numero de ventas del vendedor 5: 11
Por favor introduzca el numero de ventas del vendedor 6: 118
Por favor introduzca el numero de ventas del vendedor 7: 848
Por favor introduzca el numero de ventas del vendedor 8: 231
Por favor introduzca el numero de ventas del vendedor 9: 28
Por favor introduzca el numero de ventas del vendedor 10: 90
El ganador del concurso es quien logro vender: 848 unidades

Process returned 0 (0x0)   execution time : 11.958 s
Press any key to continue.
```

4.18 Escriba una aplicación en C++ que utilice una instrucción while y la secuencia de escape de tabulación \t para imprimir la siguiente tabla de valores:

N	10*N	100*N	1000*N
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000

```
#include<iostream>
using namespace std;
int main()
{
    int contador=1;
    cout<<"N\t "<<"N*10\t "<<"N*100\t"<<"N*1000\t"<<endl;
    while (contador <=5)
    {
        cout<<contador++<<"\t "<<contador*10<<"\t "<<contador*100<<"\t
        "<<contador*1000<<endl;;
    }
    return 0;
}
```


```
N          N*10    N*100   N*1000
1          10     100     1000
2          20     200     2000
3          30     300     3000
4          40     400     4000
5          50     500     5000

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```

4.19 Utilizando una metodología similar a la del ejercicio 4.17, encuentre los dos valores más grandes de los 10 que se introdujeron. [Nota: debe introducir cada número sólo una vez.]

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

int main ()
{
    int mayor, mayor2, numero, contador=1;
    mayor2 = mayor = INT_MIN;
    while (contador <= 10)
    {
        cout<<"Ingrese numero "<<contador++<<": ";
        cin>>numero;
        if (numero > mayor)
        {
            mayor2 = mayor;
            mayor = numero;
        }
    }
    cout<<"El mayor es: "<<mayor<<" y el segundo mayor es: "<<mayor2<<endl<<endl;
    return 0;
}
```



```
Ingrese numero 1: 5
Ingrese numero 2: 100
Ingrese numero 3: 30
Ingrese numero 4: 500
Ingrese numero 5: 3
Ingrese numero 6: 9
Ingrese numero 7: 60
Ingrese numero 8: 50
Ingrese numero 9: 10
Ingrese numero 10: 8

El mayor es: 500 y el segundo mayor es: 100

Process returned 0 (0x0)   execution time : 23.074 s
Press any key to continue.
```

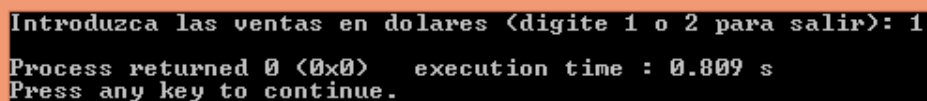
4.20 El programa de resultados de un examen de las figuras 4.16 a 4.18 asume que cualquier valor introducido por el usuario que no sea un 1 debe ser un 2. Modifique la aplicación para validar sus entradas. Para cualquier entrada, si el valor introducido es distinto de 1 o 2, debe seguir iterando hasta que el usuario introduzca un valor correcto.

a) CUANDO INGRESAMOS 1:

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

#include <iomanip>
using std::setprecision;

int main()
{
    float sueldo, salario, ventas;
    do
    {
        sueldo=200;
        ventas=0;
        salario=0;
        cout<<"Introduzca las ventas en dolares (digite 1 o 2 para salir): ";
        cin>>ventas;
        if(ventas!=1 && ventas != 2)
        {
            salario=sueldo+(ventas*0.09);
            cout << fixed << setprecision ( 2 );
            cout<<"El salario es: "<<salario<<endl<<endl;
        }
    }
    while(ventas!=1 && ventas != 2);
    return 0;
}
```



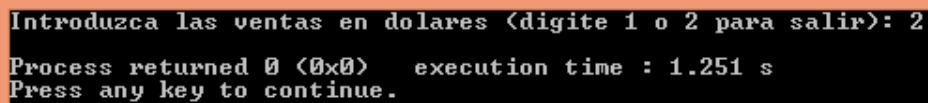
```
Introduzca las ventas en dolares (digite 1 o 2 para salir): 1
Process returned 0 (0x0) execution time : 0.809 s
Press any key to continue.
```


a) CUANDO INGRESAMOS 2:

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

#include <iomanip>
using std::setprecision;

int main()
{
    float sueldo, salario, ventas;
    do
    {
        sueldo=200;
        ventas=0;
        salario=0;
        cout<<"Introduzca las ventas en dolares (digite 1 o 2 para salir): ";
        cin>>ventas;
        if(ventas!=1 && ventas != 2)
        {
            salario=sueldo+(ventas*0.09);
            cout << fixed << setprecision ( 2 );
            cout<<"El salario es: "<<salario<<endl<<endl;
        }
    }
    while(ventas!=1 && ventas != 2);
    return 0;
}
```

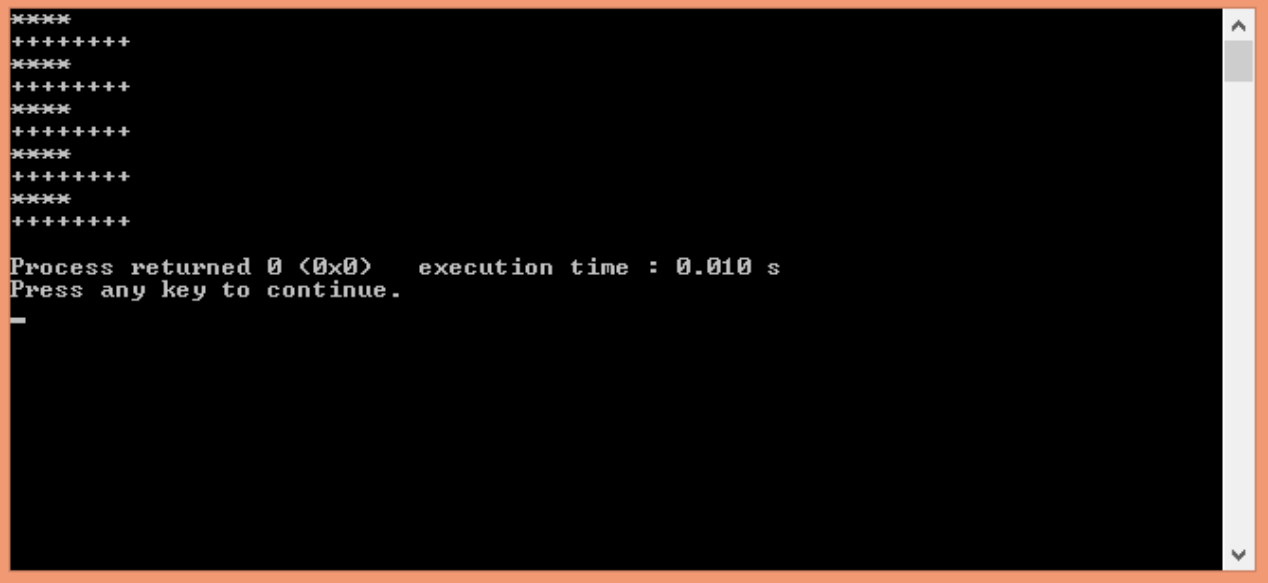


```
Introduzca las ventas en dolares (digite 1 o 2 para salir): 2
Process returned 0 (0x0)   execution time : 1.251 s
Press any key to continue.
```

4.21 ¿Qué es lo que imprime el siguiente programa?

```
1 // Ejercicio 4.21: ej04_21.cpp
2 // ¿Qué es lo que imprime este programa?
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     int cuenta = 1; // inicializa cuenta
10
11     while ( cuenta <= 10 ) // itera 10 veces
12     {
13         // imprime una línea de texto
14         cout << ( cuenta % 2 ? "****" : "++++++" ) << endl;
15         ++cuenta; // incrementa cuenta
16     } // fin de while
17
18     return 0; // indica que terminó correctamente
19 } // fin de main
```

RESPUESTA: Imprime una línea de 4 asteriscos y en la siguiente línea imprime una línea de 8 “más”, hasta que el contador cuente 10 veces.



```
****
++++++
****
++++++
****
++++++
****
++++++

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
-
```

4.22 ¿Qué es lo que imprime el siguiente programa?

```
1 // Ejercicio 4.22: ej04_22.cpp
2 // ¿Qué es lo que imprime este programa?
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main()
8 {
9     int fila = 10; // inicializa fila
10    int columna; // declara columna
11
12    while ( fila >= 1 ) // itera hasta que fila < 1
13    {
14        columna = 1; // establece columna a 1 cuando empieza la iteración
15
16        while ( columna <= 10 ) // itera 10 veces
17        {
18            cout << ( fila % 2 ? "<" : ">" ); // salida
19            ++columna; // incrementa columna
20        } // fin de while interior
21
22        --fila; // decrementa fila
23        cout << endl; // empieza nueva línea de salida
24    } // fin de while exterior
25
26    return 0; // indica que terminó correctamente
27 }
```

RESPUESTA: El programa imprime lo siguiente.

4.23 (Problema del else suelto) Determine la salida de cada uno de los siguientes conjuntos de código, cuando **x** es 9 y **y** es 11, y cuando **x** es 11 y **y** es 9. Observe que el compilador ignora la sangría en un programa en C++. El compilador de C++ siempre asocia un else con el if que le precede inmediatamente, a menos que se le indique de otra forma mediante la colocación de llaves {}. A primera vista, el programador tal vez no esté seguro de cuál if corresponde a cuál else; esta situación se conoce como el “problema del else suelto”. Hemos eliminado la sangría del siguiente código para hacer el problema más retador. [Sugerencia: aplique las convenciones de sangría que ha aprendido.]

a)

```
if ( x < 10 )
if ( y > 10 )
cout << "*****" << endl;
else
cout << "#####" << endl;
cout << "$$$$$$" << endl;
```

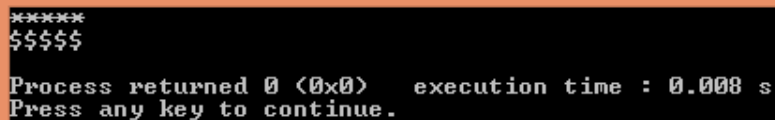
Para: x=9 y=11

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

int main()

{
    int x=9,y=11;
    if ( x < 10 )
        if ( y > 10 )
            cout << "*****" << endl;
        else
            cout << "#####" << endl;
            cout << "$$$$$$" << endl;
    return 0;
}
```

SALIDA



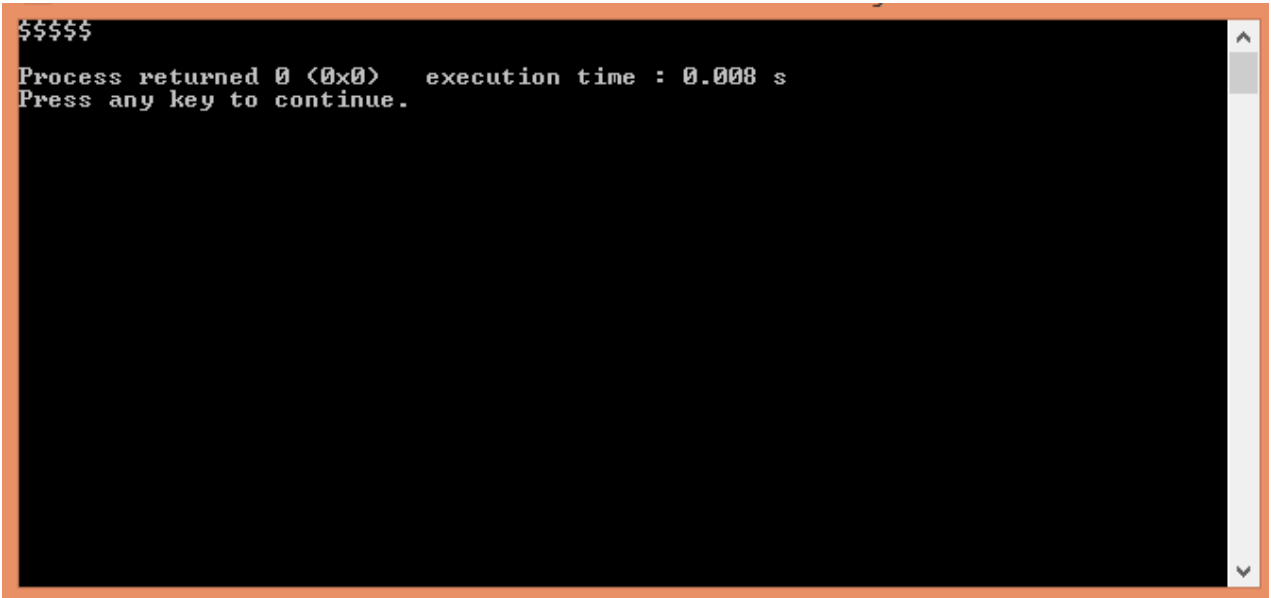
```
*****
$$$$$$

Process returned 0 (0x0)   execution time : 0.008 s
Press any key to continue.
```

Para: x=11 y=9

```
#include<iostream>
using namespace std;
int main()
{
    int x=11,y=9;
    if ( x < 10 )
        if ( y > 10 )
            cout << "*****" << endl;
        else
            cout << "#####" << endl;
            cout << "$$$$$$" << endl;
    return 0;
}
```

SALIDA



```
$$$$$$
Process returned 0 (0x0)   execution time : 0.008 s
Press any key to continue.
```

b)

```
if ( x < 10 )
{
    if ( y > 10 )
        cout << "*****" << endl;
    }
    else
    {
        cout << "#####" << endl;
        cout << "$$$$$$" << endl;
    }
}
```

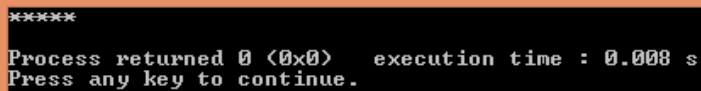
Para: x=9 y=11

```
#include<iostream>
using namespace std;
int main()

{
    int x=9,y=11;
    if ( x < 10 )
    {
        if ( y > 10 )
            cout << "*****" << endl;
        }
    else
    {
        cout << "#####" << endl;
        cout << "$$$$$$" << endl;
    }

    return 0;
}
```

SALIDA



```
*****
Process returned 0 (0x0)   execution time : 0.008 s
Press any key to continue.
_
```

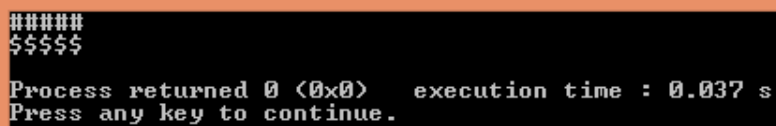
Para: x=11 y=9

```
#include<iostream>
using namespace std;
int main()

{
    int x=11,y=9;
    if ( x < 10 )
    {
        if ( y > 10 )
            cout << "*****" << endl;
    }
    else
    {
        cout << "#####" << endl;
        cout << "$$$$$$" << endl;
    }

    return 0;
}
```

SALIDA



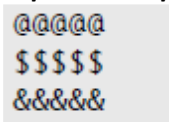
```
#####$$$$$$
Process returned 0 (0x0)   execution time : 0.037 s
Press any key to continue.
```

4.24 (Otro problema de else suelto) Modifique el siguiente código para producir la salida que se muestra. Utilice las técnicas de sangría apropiadas. No debe hacer modificaciones en el código, sólo insertar llaves o modificar la sangría del código. El compilador ignora la sangría en un programa en C++. Hemos eliminado la sangría en el código dado, para hacer el problema más retador.
[Nota: es posible que no se requieran modificaciones.]

```
if ( y == 8 )
if ( x == 5 )
cout << "@@@@@" << endl;

else
cout << "#####" << endl;
cout << "$$$$$" << endl;
cout << "&&&&" << endl;
```

a) *Suponiendo que $x = 5$ y $y = 8$, se produce la siguiente salida:*

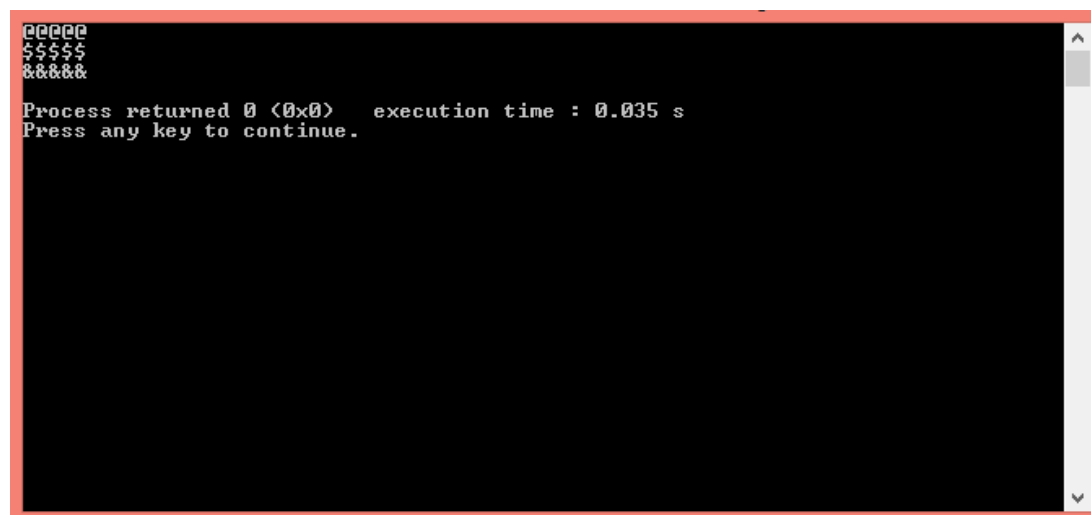


```
#include<iostream>
using namespace std;
int main()

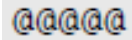
{
    int x=5,y=8;
    if ( y == 8 )
        if ( x == 5 )
            cout << "@@@@@" << endl;
        else
            cout << "#####" << endl;
            cout << "$$$$$" << endl;
            cout << "&&&&" << endl;

    return 0;
}
```

SALIDA



b) Suponiendo que $x = 5$ y $y = 8$, se produce la siguiente salida:



aaaaa

```
#include<iostream>
using namespace std;
int main()

{
    int x=5,y=8;
    if ( y == 8 )

        if ( x == 5 )
            cout << "aaaaa" << endl;
        else
        {
            cout << "#####" << endl;
            cout << "$$$$$" << endl;
            cout << "&&&&" << endl;
        }

    return 0;
}
```

SALIDA



aaaaa

Process returned 0 (0x0) execution time : 0.008 s
Press any key to continue.

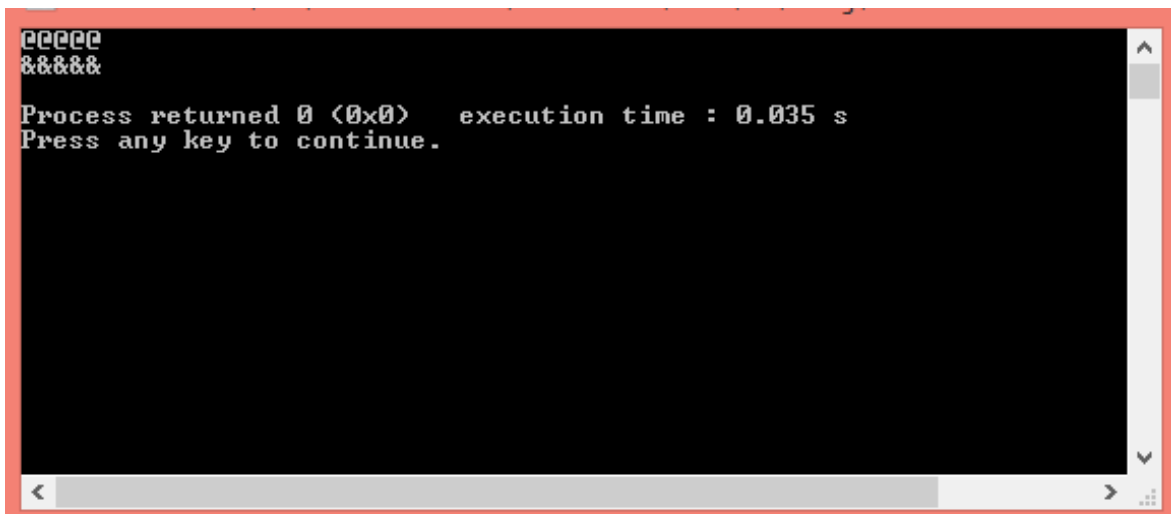
c) Suponiendo que $x = 5$ y $y = 8$, se produce la siguiente salida:

```
@@@@@
&&&&&
```

```
#include<iostream>
using namespace std;
int main()

{
    int x=5,y=8;
    if ( y == 8 )
    if ( x == 5 )
        cout << "@@@@@" << endl;
        else
        {
            cout << "#####" << endl;
            cout << "$$$$$" << endl;
        }
        cout << "&&&&&" << endl;
    return 0;
}
```

SALIDA



```
@@@@@
&&&&&

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```

- d) Suponiendo que $x = 5$ y $y = 7$, se produce la siguiente salida. [Nota: las tres últimas instrucciones de salida después del **else** forman parte de un bloque.]

```
#####  
$$$$$$  
&&&&&
```

```
#include<iostream>  
using namespace std;  
int main()  
{  
    int x=5,y=7;  
    if ( y == 8 )  
    {  
        if ( x == 5 )  
            cout << "#####" << endl;  
    }  
    else  
        cout << "#####" << endl;  
        cout << "$$$$$$" << endl;  
  
    cout << "&&&&&" << endl;  
    return 0;  
}
```

SALIDA

```
#####  
$$$$$$  
&&&&&  
  
Process returned 0 (0x0)   execution time : 0.035 s  
Press any key to continue.  
-
```

4.25 Escriba un programa que pida al usuario que introduzca el tamaño del lado de un cuadrado y que muestre un cuadrado hueco de ese tamaño, compuesto de asteriscos y espacios en blanco. Su programa debe funcionar con cuadrados que tengan lados de todas las longitudes entre 1 y 20. Por ejemplo, si su programa lee un tamaño de 5, debe imprimir:

```

* * * * *
*       *
*       *
*       *
*       *
* * * * *

```

```

#include <iostream>

using namespace std;

int main()
{
    int n;
    cout<<"Ingrese el lado del cuadrado: ";
    cin>>n;
    cout<<endl;
    for(int i=1;i<=n;i++){
        cout<<"*";
        cout<<endl;
        for(int j=1;j<=n-2;j++){
            cout<<"*";
            for(int i=1;i<=n-2;i++){
                cout<<" ";
            }
            cout<<"*"<<endl;}
        for(int i=1;i<=n;i++){
            cout<<"*";
        }
        cout<<endl;
    }

    return 0;
}

```

SALIDA

```

Ingrese el lado del cuadrado: 5
* * * * *
*       *
*       *
*       *
*       *
* * * * *

Process returned 0 (0x0)   execution time : 1.148 s
Press any key to continue.

```

```

Ingrese el lado del cuadrado: 20
*****
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*               *
*****

Process returned 0 (0x0)   execution time : 15.345 s
Press any key to continue.

```

4.26 Un palíndromo es un número o una frase de texto que se lee igual al derecho y al revés. Por ejemplo, cada uno de los siguientes enteros de cinco dígitos es un palíndromo: 12321, 55555, 45554 y 11611. Escriba una aplicación que lea un entero de cinco dígitos y determine si es un palíndromo. [Sugerencia: use los operadores de división y módulo para separar el número en sus dígitos individuales.]

```
#include <iostream> //librería
using namespace std; //librería
int main()
{
    int numero;
    int n1,n2,n3,n4,n5;
    int var;
    do
    {
        cout<<"INGRESE NUMERO ENTERO DE 5 DIGITOS: ";
        cin>>numero;
    } while(numero<0 | numero>99999);
    var=10000;
    n1=numero/var;
    numero=numero-(var*n1);
    var=var/10;
    n2=numero/var;
    numero=numero-(var*n2);
    var=var/10;
    n3=numero/var;
    numero=numero-(var*n3);
    var=var/10;
    n4=numero/var;
    n5=numero-(var*n4);
    if(n1==n5 & n2==n4)
        cout<<"\nEL NUMERO ES PALINDROMO"<<endl<<endl;
    else
        cout<<"EL NUMERO NO ES PALINDROMO"<<endl;
    return 0;
}
```

SALIDA



```
EL NUMERO ES PALINDROMO
```

```
Process returned 0 (0x0)   execution time : 3.842 s
Press any key to continue.
```

4.27 Escriba un programa que reciba como entrada un entero que contenga sólo 0s y 1s (es decir, un entero “binario”), y que imprima su equivalente decimal. Use los operadores módulo y división para elegir los dígitos del número “binario” uno a la vez, de derecha a izquierda. En forma parecida al sistema numérico decimal, donde el dígito más a la derecha tiene un valor posicional de 1 y el siguiente dígito a la izquierda tiene un valor posicional de 10, después 100, después 1000, etcétera, en el sistema numérico binario, el dígito más a la derecha tiene un valor posicional de 1, el siguiente dígito a la izquierda tiene un valor posicional de 2, luego 4, luego 8, etcétera. Así, el número decimal 234 se puede interpretar como $2 * 100 + 3 * 10 + 4 * 1$.

El equivalente decimal del número binario 1101 es $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$, o $1 + 0 + 4 + 8$, o 13. [Nota: para aprender más acerca de los números binarios, consulte el apéndice D.]

CÓDIGO C++

```
#include <iostream>
#include <stdio.h>
#define Tamano 35
using namespace std;
void Recibe( int a[]);
void Imprime( int b[]);

int main()
{
    int arreglo[Tamano];
    int evaluar;
    cout<<"\nEste programa recibe un numero binario y lo convierte a decimal"<<endl;
    Recibe(arreglo);
    Imprime(arreglo);
    return 0;
}

void Recibe( int a[])
{
    int i = 0;
    for( i = 0; i < Tamano; i++ )
        a[i] = ' ';
    cout<<"\nIntroduzca un numero binario: ";
    i = 0;
    while ( (a[i++] = getchar()) != '\n' && Tamano > i );
    cout<<endl;
}

void Imprime( int b[] )
{
    int i;
    int potencia = 1;
    int decimal = 0;
    int inicio;

    for ( i = Tamano - 1; i >= 0; i-- )
    {
        if ( ' ' != b[i] )
        {
            inicio = i;
            break;
        }
    }
```

```

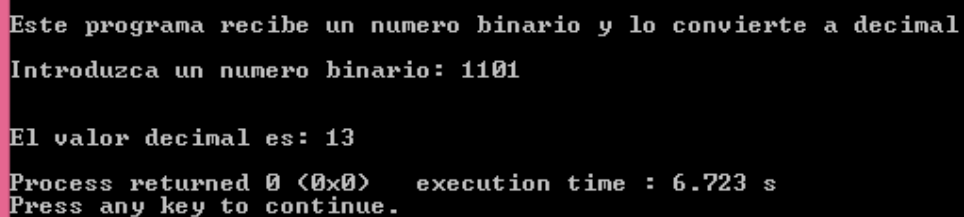
    }
    int valido = 1;

    for ( i = inicio; i >= 0; i-- )
    {
        switch( b[i] )
        {
            case '0':
                decimal += 0*potencia;
                potencia *= 2;
                break;
            case '1':
                decimal += 1*potencia;
                potencia *= 2;
                break;
            default:
                if ( '\n' != b[i])
                {
                    cout<<"\nERROR. LA CADENA NO ES VALIDA!\n";
                    valido = 0;
                    cout<<"\nEste caracter no es valido: ";
                    putchar(b[i]);
                    cout<<endl;
                }
                break;
        }
    }

    if ( 1 == valido )
        cout<<"\nEl valor decimal es: "<<decimal<<endl;
    else
        cout<<"\nEl numero introducido no es valido."<<endl;
}

```

SALIDA



```

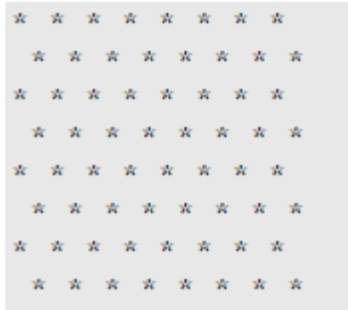
Este programa recibe un numero binario y lo convierte a decimal
Introduzca un numero binario: 1101

El valor decimal es: 13
Process returned 0 (0x0) execution time : 6.723 s
Press any key to continue.

```

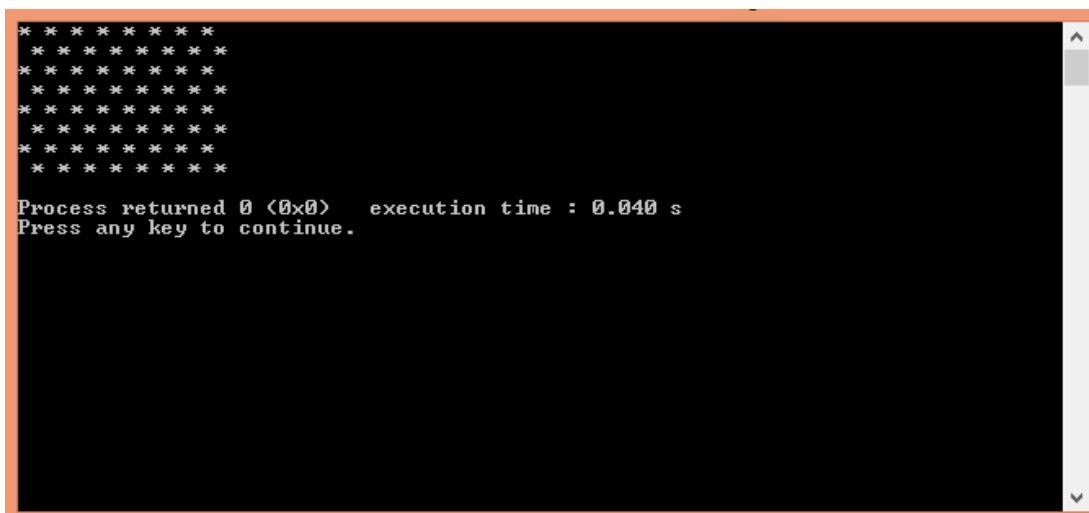
4.28 Escriba un programa que muestre el patrón de tablero de damas que se muestra a continuación. Su programa debe utilizar sólo tres instrucciones de salida, una para cada una de las siguientes formas:

```
cout << "*" << " ";  
cout << ' ' << " ";  
cout << endl;
```



CÓDIGO

```
#include <iostream>  
using namespace std;  
int main()  
{  
    for(int i= 1; i <= 8; i++)  
    {  
        for(int j = 1; j <= 16; j++)  
        {  
            if ((i + j)%2 == 0)  
            {  
                cout << "*";  
            }  
            else  
            {  
                cout << " ";  
            }  
        }  
        cout << endl;  
    }  
    return 0; }
```



4.29 Escriba un programa que imprima las potencias del entero 2; a saber, 2, 4, 8, 16, 32, 64, etcétera. Su ciclo while no debe terminar (es decir, debe crear un ciclo infinito). Para ello, simplemente use la palabra clave true como la expresión para la instrucción while. ¿Qué ocurre cuando ejecuta este programa?

CÓDIGO

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

int main()
{
    int y=2;
    double z=0;
    int contador=1;

    do
    {
        z=pow(y,contador++);
        cout<<z<<endl;
    }while ( true );
    return 0;
}
```

SALIDA

inf

4.30 Escriba un programa que lea el radio de un círculo (como un valor double), calcule e imprima el diámetro, la circunferencia y el área. Use el valor 3.14159 para π .

CÓDIGO

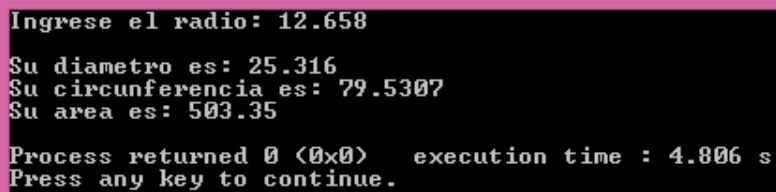
```
#include <iostream>

using namespace std;

double radio=0;
double circunferencia=0, area=0,diametro=0,pi=3.141519;

int main()
{
    cout << "Ingrese el radio: ";
    cin >> radio;
    diametro=radio*2;
    circunferencia = 2 * pi * radio;
    area = pi * radio * radio ;
    cout << "\nSu diametro es: " << diametro << endl;
    cout << "Su circunferencia es: " << circunferencia << endl;
    cout << "Su area es: " << area << endl;
    return 0;
}
```

SALIDA

A screenshot of a terminal window with a black background and white text. The text shows the program's execution: it prompts for a radius, receives the input '12.658', and then outputs the calculated diameter (25.316), circumference (79.5307), and area (503.35). At the bottom, it shows 'Process returned 0 (0x0)' and 'execution time : 4.806 s', followed by a prompt to press any key to continue. A vertical scrollbar is visible on the right side of the terminal window.

```
Ingrese el radio: 12.658
Su diametro es: 25.316
Su circunferencia es: 79.5307
Su area es: 503.35

Process returned 0 (0x0)   execution time : 4.806 s
Press any key to continue.
```

4.31 ¿Qué está mal con la siguiente instrucción? Proporcione la instrucción correcta para realizar lo que probablemente el programador trataba de hacer.

```
cout << ++( x + y );
```

RESPUESTA: El error radica en que las variables no han sido previamente declaradas ni inicializadas, además de que primero se debería efectuar la suma, y “guardarla” en otra variable, y es a esta variable “suma” a la cual se le debería realizar el incremento.

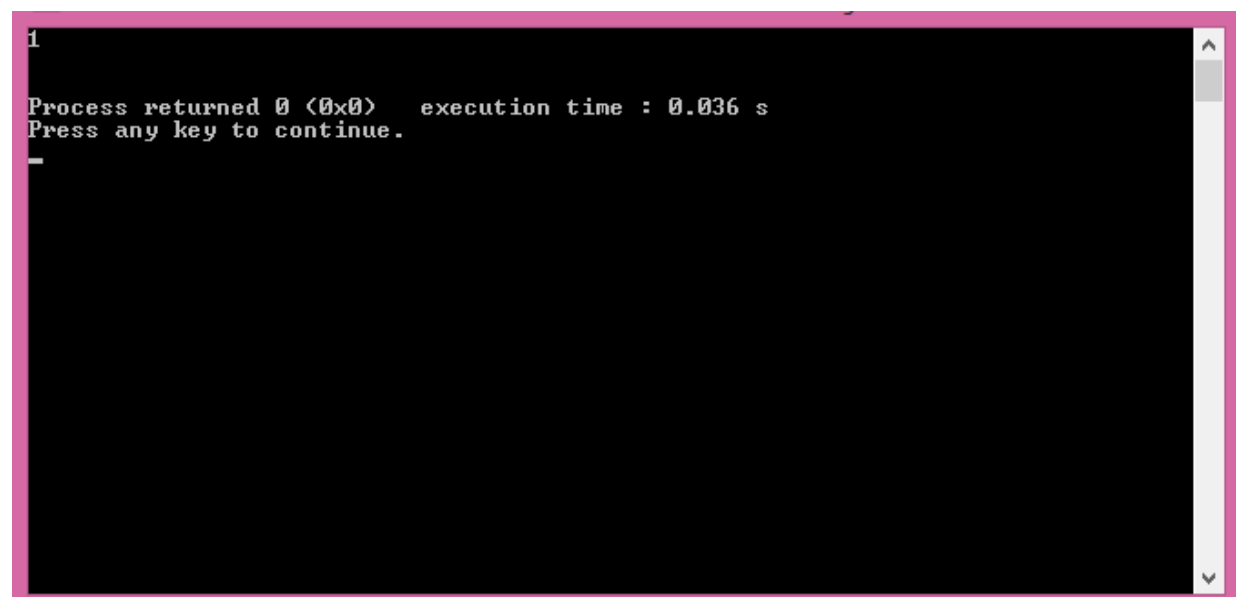
CÓDIGO

```
#include <iostream>
using namespace std;
int x=0,y=0,suma=0;

int main()
{
    suma=x+y;
    cout<< ++( suma )<<endl<<endl;

    return 0;
}
```

SALIDA



```
1
Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

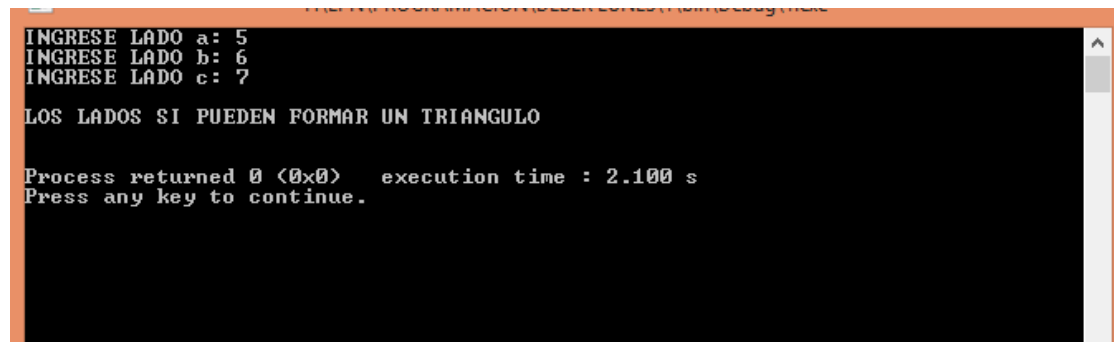
4.32 Escriba un programa que lea tres valores **double** distintos de cero, y que determine e imprima si podrían representar los lados de un triángulo.

CÓDIGO

```
#include <iostream> //librería
using namespace std;
int main()
{
    double a, b, c;
    do
    {
        cout<<"INGRESE LADO a: ";
        cin>>a;
        cout<<"INGRESE LADO b: ";
        cin>>b;
        cout<<"INGRESE LADO c: ";
        cin>>c;
        if((a<=0) || (b<=0) || (c<=0))
            cout<<"\nLOS LADOS NO FORMAN UN TRIANGULO"<<endl<<endl;
        else
            cout<<"\nLOS LADOS SI PUEDEN FORMAR UN TRIANGULO"<<endl<<endl;
    }while((a<=0) || (b<=0) || (c<=0));

    return 0;
}
```

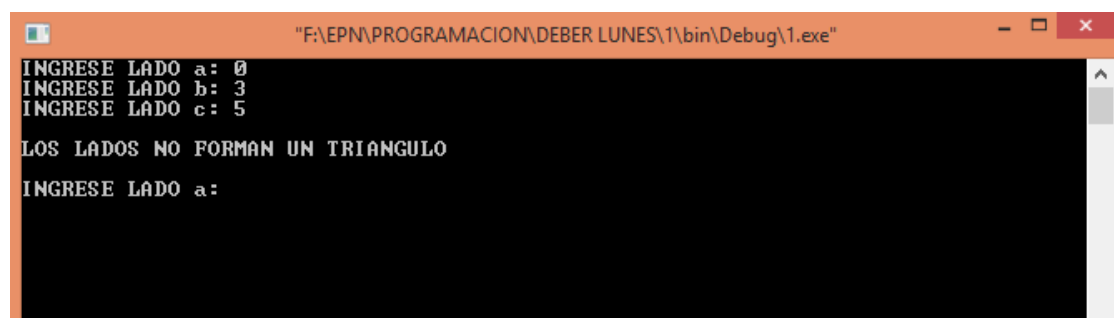
SALIDA



```
INGRESE LADO a: 5
INGRESE LADO b: 6
INGRESE LADO c: 7

LOS LADOS SI PUEDEN FORMAR UN TRIANGULO

Process returned 0 (0x0)   execution time : 2.100 s
Press any key to continue.
```



```
"F:\EPN\PROGRAMACION\DEBER LUNES\1\bin\Debug\1.exe"
INGRESE LADO a: 0
INGRESE LADO b: 3
INGRESE LADO c: 5

LOS LADOS NO FORMAN UN TRIANGULO

INGRESE LADO a:
```

4.33 Escriba un programa que lea tres enteros distintos de cero, y que determine e imprima si podrían ser los lados de un triángulo recto.

CÓDIGO

```
#include <iostream>
using namespace std;
#include <stdio.h>
int main()
{
    double lado1,lado2,lado3,auxiliar;
    cout<<"\nPrimer lado : ";
    cin>>lado1;
    cout<<"\nSegundo lado : ";
    cin>>lado2;
    cout<<"\nTercer lado : ";
    cin>>lado3;

    if (lado1>lado2)
    {
        auxiliar=lado1;
        lado1=lado2;
        lado2=auxiliar;
    }
    if (lado3<lado1)
    {
        auxiliar=lado3;
        lado3=lado2;
        lado2=lado1;
        lado1=auxiliar;
    }
    if (lado3<lado2)
    {
        auxiliar=lado2;
        lado2=lado3;
        lado3=auxiliar;
    }

    cout<<"\n lado 1: "<<lado1<<endl;
    cout<<"\n lado 2: "<<lado2<<endl;
    cout<<"\n lado 3: "<<lado3<<endl;

    if (lado3>=lado1+lado2){
        cout<<"\nEsto no es un triangulo"<<endl;
        cout<<"\nPOR LO TANTO MUCHO MENOS PUEDE SER TRIANGULO RECTO"<<endl;}
    else
        if ((lado1==lado2) && (lado2==lado3)){
            cout<<"\nTriangulo Equilatero"<<endl;
            cout<<"\nNO PUEDE SER TRIANGULO RECTO"<<endl;}

    else
        if ((lado1==lado2) || (lado1==lado3) || (lado2==lado3)){
            cout<<"\nTriangulo Isosceles"<<endl<<endl;
            cout<<"\nADEMAS HAY POSIBILIDAD DE QUE SEA UN TRIANGULO RECTO"<<endl;}
```

```

else{
    cout<<"\nTriangulo Escaleno"<<endl;
    if ((lado3<lado1+lado2)&&(lado3*lado3==lado1*lado1+lado2*lado2))
        cout<<"\nY ADEMAS SI ES UN TRIANGULO RECTO"<<endl;}

    return 0;
}

```

SALIDA

```

Primer lado : 3
Segundo lado : 4
Tercer lado : 5
    lado 1: 3
    lado 2: 4
    lado 3: 5
Triangulo Escaleno
Y ADEMAS SI ES UN TRIANGULO RECTO
Process returned 0 (0x0)   execution time : 2.216 s
Press any key to continue.
_

```

```

Primer lado : 3
Segundo lado : 3
Tercer lado : 3
    lado 1: 3
    lado 2: 3
    lado 3: 3
Triangulo Equilatero
NO PUEDE SER TRIANGULO RECTO
Process returned 0 (0x0)   execution time : 1.906 s
Press any key to continue.
_

```

4.34 (Criptografía) Una compañía desea transmitir datos a través del teléfono, pero le preocupa que sus teléfonos puedan estar intervenidos. Todos los datos se transmiten como enteros de cuatro dígitos. La compañía le ha pedido a usted que escriba un programa que cifre sus datos, de manera que éstos puedan transmitirse con más seguridad. Su programa debe leer un entero de cuatro dígitos introducido por el usuario y cifrarlo de la siguiente manera: reemplace cada dígito con (el resultado de sumar 7 al dígito) módulo 10. Luego intercambie el primer dígito con el tercero, e intercambie el segundo dígito con el cuarto. Después imprima el entero cifrado. Escriba un programa separado que reciba como entrada un entero de cuatro dígitos cifrado, y que lo descifre para formar el número original.

CÓDIGO QUE CIFRA:

```
#include <iostream>
using std::cout;
using std::cin;
using std::endl;

int main()
{
    int digito1;
    int digito2;
    int digito3;
    int digito4;
    int numero;
    int resto;

    cout << "Ingrese el numero a cifrar:\t";
    cin >> numero;
    cout << endl;

    digito1= numero / 1000;
    resto= numero % 1000;
    digito2= resto / 100;
    resto = resto % 100;
    digito3= resto / 10;
    resto= resto % 10;
    digito4= resto;

    digito1= ((digito1 + 7) % 10);
    digito2= ((digito2 + 7) % 10);
    digito3= ((digito3 + 7) % 10);
    digito4= ((digito4 + 7) % 10);

    cout << "Numero cifrado:\t";
    cout << digito3 << digito4 << digito1 << digito2 << endl;
    cin.get();
    cin.get();
    return 0;
}
```

CÓDIGO QUE DECIFRA:

```
#include <iostream>
using std::cout;
using std::endl;
using std::cin;

int main()
{
    int digito1;
    int digito2;
    int digito3;
    int digito4;
    int numero;
    int resto;

    cout << "Numero a descrifrar:\t";
    cin >> numero;
    cout << endl;

    digito1= numero / 1000;
    resto= numero % 1000;
    digito2= resto / 100;
    resto= resto % 100;
    digito3= resto / 10;
    resto = resto % 10;
    digito4= resto;

    digito2= digito2 - 7;
    digito1= digito1 + 3;
    digito3= digito3 + 3;
    digito4= digito4 + 3;

    cout <<"Numero descifrado: \t";
    cout <<digito3 << digito4 << digito1 << digito2 << endl;
    cin.get();
    cin.get();

    return 0;
}
```


SALIDA CIFRADO:

```
Ingrese el numero a cifrar: 5555
Numero cifrado: 2222
```

SALIDA DECIFRADO:

```
Numero a descifrar: 2222
Numero descifrado: 5555
```

4.35 El factorial de un entero n no negativo se escribe como $n!$ (n factorial) y se define de la siguiente manera:

$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 1$ (para valores de n mayores o iguales a 1)

y

$n! = 1$ (para $n = 0$ o $n = 1$).

Por ejemplo, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, que es 120. Use instrucciones while en cada uno de los siguientes casos:

a) Escriba una aplicación que lea un entero no negativo, que calcule e imprima su factorial.

CÓDIGO

```
#include<iostream>
using namespace std;
int factorial(int);

int main()
{
    int n;
    cout<<"Ingrese un numero: ";
    cin>>n;
    cout<<"El factorial del numero es: "<<factorial(n);
    return 0;
}

int factorial(int n)
{
    int fact;
    if (n==0)
        fact=1;
    else
    {
        for(int i=1;i<=n;i++)
        {
            fact=1;
            for(int j=1;j<=i;j++)
            {
                fact=fact*j;
            }
        }
    }
    return fact;
}
```

SALIDA

```
Ingrese un numero: 5
El factorial del numero es: 120
Process returned 0 (0x0)   execution time : 1.221 s
Press any key to continue.
```

b) Escriba un programa que estime el valor de la constante matemática e, utilizando la fórmula:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

CÓDIGO

```
#include <stdio.h>
#include <math.h>
#include <iostream>
using namespace std;
float factorial (int num)
{
    if (num==0)
        return 1;
    else
        return num * factorial(num-1);
}

double e(int n)
{
    double sum = 0;
    for(int i = 0; i < n; i++){
        sum += 1/factorial(i);
    }
    return sum;
}

int i=0;
float X, S=0, termino=1;
```

```

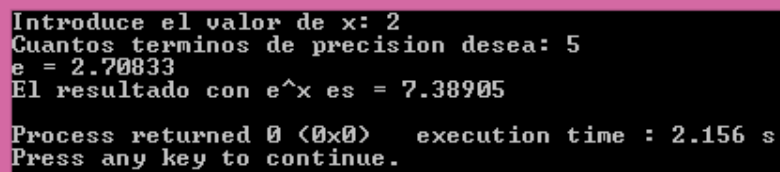
int main()
{

    cout<<"Introduce el valor de x: ";
    cin>>X;
    while(termino>=0.0001)
    {
        termino=pow(X,i)/factorial(i);
        S+= termino;
        i++;
    }
    int n;
        cout <<"Cuantos terminos de precision desea: ";
        cin >> n;
        cout<<"e = "<<e(n);
    cout<<"\nEl resultado con e^x es = "<<S<<endl;


    return 0;
}

```

SALIDA



Introduce el valor de x: 2
 Cuantos terminos de precision desea: 5
 e = 2.70833
 El resultado con e^x es = 7.38905
 Process returned 0 (0x0) execution time : 2.156 s
 Press any key to continue.

c) Escriba una aplicación que calcule el valor de e^x , utilizando la fórmula

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

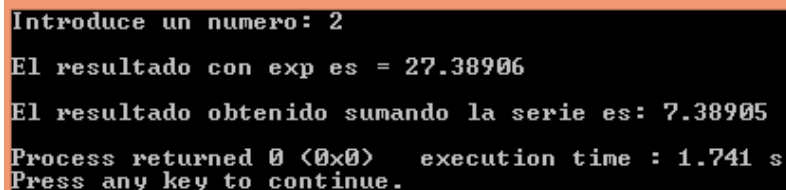
CÓDIGO

```
#include <stdio.h>
#include <math.h>
#include <iostream>
using namespace std;
float factorial (int num) {
    if (num==0)
        return 1;
    else
        return num * factorial(num-1);
}
int i=0;
float X, S=0, termino=1;
int main(){

    cout<<"Introduce un numero: ";
    cin>>X
    while(termino>=0.0001){
        termino=pow(X,i)/factorial(i);
        S+= termino;
        i++;
    }
    cout<<"\nEl resultado con exp es = "<<X<<exp(X)<<endl;
    cout<<"\nEl resultado obtenido sumando la serie es: "<<S<<endl;

    return 0;
}
```

SALIDA



```
Introduce un numero: 2
El resultado con exp es = 27.38906
El resultado obtenido sumando la serie es: 7.38905
Process returned 0 (0x0)   execution time : 1.741 s
Press any key to continue.
_
```

4.36 [Nota: este ejercicio corresponde a la sección 4.13, una parte de nuestro Ejemplo práctico de Ingeniería de Software.]

Describe en 200 palabras o menos qué es un automóvil y qué hace. Enliste los sustantivos y verbos por separado. En el texto, indicamos que cada sustantivo podría corresponder a un objeto que habrá que construir para implementar un sistema, en este caso un auto. Elija cinco de los objetos que enlistó y, para cada uno, enliste varios atributos y comportamientos. Describa brevemente cómo interactúan estos objetos entre sí, y con los demás objetos en su descripción. Acaba de realizar varios de los pasos clave en un típico diseño orientado a objetos.

DESCRIPCIÓN

encender

comportamiento:

- encender motor del auto
- encender luces
- consumir combustible
- hacer ruido del motor

atributos

- sin tardar
- tardar

apagar

comportamiento:

- apagar luces
- dejar de hacer ruido
- apagar motor del auto

atributos:

- sin tardar

acelerar

comportamiento:

- aumentar velocidad

atributos:

- hasta llegar a 30 km/h tardando 3 segundos
- hasta llegar a 50 km/h tardando 2 segundos

desacelerar

comportamiento:

- reducir velocidad

atributos:

- bajar velocidad hasta 30 km/h tardando 3 segundos
- bajar velocidad hasta 50 km/h tardando 2 segundos

frenar:

comportamiento:

- reducir velocidad a 0

atributos:

- hacer ruido
- prender luces rojas