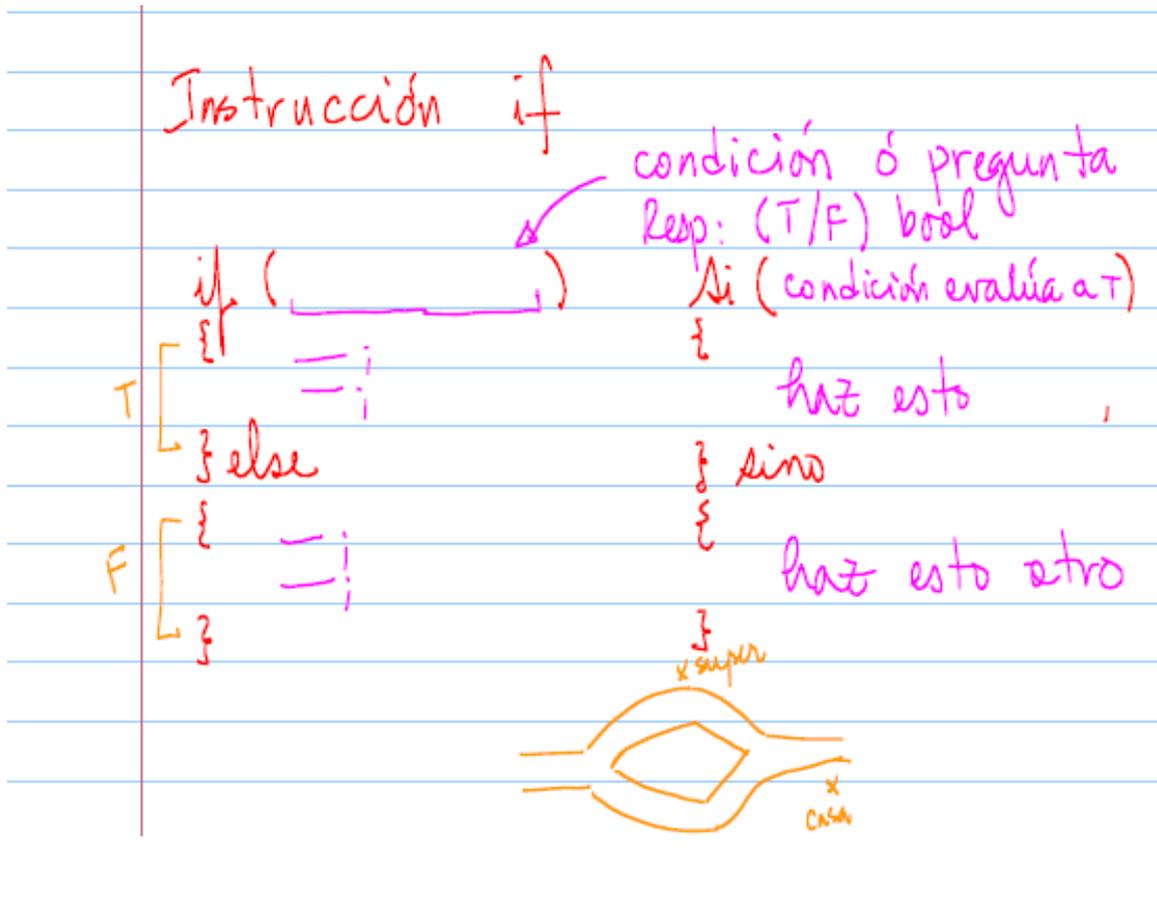


# Clase10ene2013

Tuesday, January 08, 2013  
6:28 PM



Código fuente hecho con notepad ++

```
C:\MinGW\bin\ifproblem.cpp - Notepad++  
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?  
MICROSOFT_WINDOWS_XP_PROFES.TXT | tareas.txt | ITodos.txt | license.txt  
selectall600.iim | grades.txt | gradesKey.cpp | ifproblem.cpp | perimeter.cpp  
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 int main() {  
6     bool state;  
7     cout << "Ejemplo de if statement. Enter state: ";  
8     cin >> state;  
9     if(state) {  
10        cout << "state variable was TRUE." << endl;  
11    }else{  
12        cout << "state variable was FALSE." << endl;  
13    }  
14    return 0;  
15 }
```

length : 276 lines : 15 | Ln : 8 Col : 18 Sel : 0 | Dos\Windows | ANSI | INS

Compilando y ejecutando:

```
Administrator: C:\Windows\system32\cmd.exe
C:\MinGW\bin>g++ ifproblem.cpp -o ifproblem.exe
C:\MinGW\bin>ifproblem.exe
Ejemplo de if statement. Enter state: 1
state variable was TRUE.
C:\MinGW\bin>ifproblem.exe
Ejemplo de if statement. Enter state: 0
state variable was FALSE.
C:\MinGW\bin>
```

Ejercicio:

- en una semana por 1 empl.
- Si las horas trabajadas son mayores que 40 entonces imprima "trabajo horas extras"; sino imprima "no trabajo horas extras".

#### FASE 1: Analisis

- Escribir mensaje con instrucciones
  - o "Por favor entre sus horas trabajadas durante esta semana: ";
- La computadora lee el dato que entra el usuario
  - o Leer en una variable entera ejemplo int horasTrabajadas;
- Decidir si el dato entrado es mayor que cuarenta y en ese caso escribir:
  - o "El empleado trabajó horas extras"
- En caso contrario escribir:
  - o "El empleado no trabajó horas extras"
- Return;

#### FASE 2: Programa

```

12  #include <iostream>
13  using namespace std;
14  const int HORASSEMANA = 40;
15
16  int main() {
17      int horasTrabajadas;
18      int horasExtra;
19
20      cout << "Por favor entre sus horas trabajadas durante esta semana: ";
21      cin >> horasTrabajadas;
22
23      cout << "Las horas trabajadas son: " << horasTrabajadas << endl;
24
25      if(horasTrabajadas > HORASSEMANA) {
26          cout << "El empleado trabajo horas extras: ";
27          horasExtra = horasTrabajadas - HORASSEMANA;
28          cout << horasExtra << endl;
29      }else{
30          cout << "El empleado no trabajo horas extras" << endl;
31      }
32      return 0;
33  }

```

FASE 3: verificar

```

C:\MinGW\bin>g++ horas.cpp -o horas.exe
C:\MinGW\bin>horas.exe
Por favor entre sus horas trabajadas durante esta semana: 43
Las horas trabajadas son: 43
El empleado trabajo horas extras: 3
C:\MinGW\bin>_

```

```
double hrs;
```

```
cout << "Entre las horas trabajadas en la"  
      << "semana por el empleado:";  
cin >> hrs;
```

```
if ( hrs > 40.0 )
```

```
{  
  cout << "Trabajo horas extras\n\n";
```

```
} else
```

```
{  
  cout << "No trabajo horas extras\n\n";
```

```
}
```

Otra forma alternativa:

```
if ( hrs <= 40.0 )
```

```
{  
  cout << "No trabajo horas extras\n\n";
```

```
} else
```

```
{  
  cout << "Trabajo horas extras\n\n";
```

```
}
```

## PROBLEMA:

Una compañía paga a sus empleados por las horas trabajadas en la semana. Las horas extras las paga al doble del pago por hora regular. Escriba un prog. en C++ que permita calcular e imprimir la cantidad a pagar en una semana a un empleado que cobra por horas.

⊛ Una jornada de trabajo regular tiene 40 horas a la semana.

El usuario debe entrar tanto las horas trabajadas (ya esta) como el salario por hora (hacer). La variable para almacenar el salario por hora es **int salHora;**

El programa debe calcular el total del salario semanal:

$\text{salarioSemanal} = \text{salHora} * \text{horasTrabajadas};$  <- solamente si las horasTrabajadas son menos de cuarenta.

$\text{salarioSemanal} = \text{salHora} * \text{HORASEMANA} + 2 * \text{salHora} * \text{horasExtra};$

## FASE 1: Analisis

Entradas      Procesamiento      Salidas

## FASE 2: PROGRAMA

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

```
① int main()  
{
```

## FASE 3: TEST Ó PRUEBA MANUAL DE LA SOL.

### Validación de los datos de entrada:

Es el proceso de verificar, en la medida de lo posible, que un dato que se solicita al usuario, se entre correctamente antes de proceder a trabajar con él en la solución del problema.

- Ej.:
- Un precio debe ser cant. positiva ó cero (gratis!)
  - Cant. de estudiantes no debe ser negativo.
  - Una nota debe ser una cant. entre 0 y 100 etc.

Si al ejercicio anterior se le incorporara la validación de los datos de entrada (o sea, la verificación, antes de proceder a solucionar el problema, de que las horas trabajadas por el empleado en la semana y su pago por hora NO sean negativos)

```
//Input verification
if(horasTrabajadas < 0) {
    cout << "Por favor entre un numero positivo: ";
    cin >> horasTrabajadas;
}
```

## BOOLEAN VARIABLES

/\* Un banco aprueba una transacción si el cliente presenta una licencia válida \*/

```
bool lic;
```

```
cout << "Entre 1 si el cliente presentó una lic. válida"
      << " o entre 0 si no lo presentó: ";
```

```
cin >> lic;
```

FORMA 2:

```
FORMA 1: if ( lic == 1 ) // si cliente tiene lic. válida
```

```
{
    cout << "Se aprueba la transacción\n\n";
} else
```

```
{
    cout << "No se aprueba la transacción\n\n";
}
```

ó **!(lic == 1) // FORMA 5**  
 ! lic  
 FORMA 4:  
 FORMA 3: **if ( lic == 0 ) // si cliente no tiene lic. válida**  
 {  
 T cout << "No se aprueba la transacción\n\n";  
 } else  
 {  
 F cout << "Se aprueba la transacción\n\n";  
 }

## BOOLEAN AND &&

Ej: Otro banco aprueba la transacción si  
 el cliente presenta lic. válida y alguna  
 otra id. con foto.

```
bool lic, id;
```

```
cout << "Entre 1 si el cliente presento una lic. valida"  
      << " o entre 0 si no la presento: ";
```

```
cin >> lic;
```

```
cout << "Entre 1 si el cliente presento otra id. con foto"  
      << " o entre 0 si no la presento: ";
```

```
cin >> id;
```

```
lic && id  
if ( lic == 1 && id == 1 )  
{
```

```
    → cout << "Se aprueba la transaccion\n";
```

```
} else  
{
```

```
    → cout << "No se aprueba la transaccion\n";
```

```
}
```

## BOOLEAN OR ||

Ej: Otro banco aprueba la transacción si el cliente presenta lic. válida o alguna otra id. con foto.

```
bool lic, id;
```

```
cout << "Entre 1 si el cliente presento una lic. valida"  
      << " o entre 0 si no la presento: ";
```

```
cin >> lic;
```

```
cout << "Entre 1 si el cliente presento otra id. con foto"  
      << " o entre 0 si no la presento: ";
```

```
cin >> id;
```

lic || id

```
if ( lic == 1 || id == 1 )  
{
```

```
    → cout << "Se aprueba la transaccion\n\n";
```

```
} else
```

```
{
```

```
    → cout << "No se aprueba la transaccion\n\n";
```

```
}
```

### Ejercicio

/\* Un promedio final es válido si está entre  
cero y 100. \*/

```
double pf;
```

```
cout << "Entre el prom. final: ";  
cin >> pf;
```

```
// Si el pf está entre 0 y 100
```

```
X if (0 <= pf <= 100 )
```

```
    {  
      cout << "Prom. valido\n\n";  
    } else
```

```
    {  
      cout << "Prom. inv.\n\n";  
    }
```

-5.0 97.0

Mal expresado en C++

Siempre da true → true T<sup>(2)</sup> X

por lo que no está bien expresada la condición. la sigt. expresión si funciona:

```
if ( pf >= 0.0 && pf <= 100.0 )
```

Diagram illustrating the evaluation of the condition  $pf \geq 0.0 \ \&\& \ pf \leq 100.0$  for different values of  $pf$ :

- For  $pf = -5.0$ :
  - Condition 1:  $pf \geq 0.0$  is **T** (True).
  - Condition 2:  $pf \leq 100.0$  is **T** (True).
  - Overall result: **T** (True).
- For  $pf = 97.0$ :
  - Condition 1:  $pf \geq 0.0$  is **T** (True).
  - Condition 2:  $pf \leq 100.0$  is **T** (True).
  - Overall result: **T** (True).
- For  $pf = -5.0$  (repeated):
  - Condition 1:  $pf \geq 0.0$  is **F** (False).
  - Condition 2:  $pf \leq 100.0$  is **T** (True).
  - Overall result: **F** (False).