

Prueba de Ejecución # 1

Nombre:

Cédula:

OBSERVACIONES GENERALES:

- Este examen se debe trabajar de manera estrictamente individual.
- El tiempo disponible para realizar el examen es de 3 horas, estudiantes con autorización formal de tiempo adicional lo tendrán después de transcurridas este tiempo.
- La respuesta al examen debe entregarse manuscrita en papel junto con el enunciado del examen.
- No se aceptan reclamos posteriores si el examen se escribe a lápiz y no con lapicero.
- Trabaje en forma clara y ordenada.
- Cuide su examen pues, de detectarse códigos iguales, se calificará con cero a los estudiantes relacionados.
- Se evalúa la sintaxis, la lógica del algoritmo, la eficiencia y el procedimiento.
- Apague su celular y colóquelo boca abajo en el borde de la mesa, lugar en el cual deberá permanecer durante todo el examen.
- Coloque los monitores de las computadoras del laboratorio en la posición que indique su profesor.
- Queda totalmente prohibido durante el horario de aplicación del examen encender la computadora de su estación de trabajo o utilizar cualquier otro dispositivo digital como laptop, tableta, celular, calculadora o reloj digital.
- Si requiere ir al baño, acuda antes de iniciar este examen.

I Parte: Complete (25 pts)

1) ¿Qué valor se escribe con cada cout? (12.5 pts.)

```
int main (int argc, char *argv[]) {  
    int i =2;  
    int j = 3;  
    cout << ( ( 2 * i <= j) && (i * i !=100) ) << endl; R/ _____ (1.25 pts.)  
    cout << ( ( j - 2 < j ) && ( i * 2 > j ) ) << endl; R/ _____ (1.25 pts.)  
    cout << ! (i==2) << endl; R/ _____ (1.25 pts.)  
    cout << ( (i > 13) || ( i<25) && ( j != 50) ) << endl; R/ _____ (1.25 pts.)  
    cout << ((i <= 4) || (j >= 5)) << endl; R/ _____ (1.25 pts.)  
    cout << ((i % 2 != 0) && (j / 3 != 6)) << endl; R/ _____ (1.25 pts.)  
    cout << ((j - 7 < j) && (i * 4 > j)) << endl; R/ _____ (1.25 pts.)  
    cout << !(i == 4) << endl; R/ _____ (1.25 pts.)  
    cout << ((i + 15 < 10) || (i - 8 < 30) && (j != 40)) << endl; R/ _____ (1.25 pts.)  
    cout << ((i * 2 >= 6) || (j + 3 >= 9)) << endl; R/ _____ (1.25 pts.)  
}
```

2) ¿Qué valor se escribe con cada cout? (8 pts.)

```
int calcular (int &x, int &y, int z){  
    x++;  
    y++;  
    z= x + y++;  
    cout<<z<<endl;           R/ _____ (2 pts)  
}  
  
int main (int argc, char *argv[]){  
    int a=10;  
    int b=10;  
    int c=10;  
    calcular(a,b,c);  
    cout<<a<<endl;           R/ _____ (2 pts.)  
    cout<<b<<endl;           R/ _____ (2 pts.)  
    cout<<c<<endl;           R/ _____ (2 pts.)  
  
    return 0;  
}
```

3) ¿Qué valor se escribe con cada cout? (4.5 pts.)

```
int main (int argc, char *argv[]){  
    int x, y, z;  
    x = 4;  
    y= (++x) *3;  
    z=1;  
    while (x<=5){  
        z= (x%y) + (y++) + x;  
        x++;  
    }  
    cout<<"x: "<<x <<endl;     R/ _____ (1.5 pts.)  
    cout<<"y: "<<y <<endl;     R/ _____ (1.5 pts.)  
    cout<<"z: "<<z <<endl;     R/ _____ (1.5 pts.)  
  
    return 0;  
}
```

II Parte: Desarrollo (35 pts.)

Una empresa de tecnologías de la información (TI) planea calcular el salario o pago de sus colaboradores para los diferentes puestos:

- **oficinistas**
- **programadores**
- **jefaturas**
- **consultores** (*estos últimos no están en planilla pues no son considerados propiamente empleados de la empresa, sino que trabajan bajo la modalidad de servicios profesionales*).

Scriba un programa en C++ que permita registrar a cada trabajador y calcular su salario o pago correspondiente. Para cada trabajador se requiere solicitar: **nombre, puesto, años de experiencia y horas laboradas**. Todos los puestos son contratados para alguna de las siguientes jornadas:

- **Tiempo completo (48 horas)**
- **Medio tiempo (24 horas)**
- **Cuarto de tiempo (12 horas)**
- **Por horas (n horas)**, en este rubro trabajan únicamente los **consultores**, quienes no son empleados de la empresa propiamente, sino que trabajan por servicios profesionales, son contratados en forma eventual y se les paga por hora trabajada.

Escriba las siguientes funciones:

2.1. Función calculoSalarioBase (8 pts.)

Esta función debe recibir como parámetros el **puesto** y la cantidad de **horas** laboradas por un trabajador (*48, 24, 12 o la cantidad de horas laboradas por un consultor*). La función debe retornar el **salario base** o pago del trabajador según las tablas que se presentan a continuación.

Considere que la **Tabla de Salario Base** muestra el salario base del puesto para una jornada de tiempo. No obstante, el trabajador puede estar nombrado en otras jornadas (medio tiempo o cuarto de tiempo), por lo tanto, el salario base del trabajador se debe calcular proporcional a su jornada.

Tome en cuenta que los **consultores** no reciben un salario base, sino que su pago se calcula con base en las horas laboradas. El monto de pago por hora de **consultor** se muestra en la tabla de la derecha.

Tabla de Salario Base (Empleados en planilla)	
Puesto	Salario base (para Tiempo Completo)
Oficinista	€520.000
Programador	€1.800.000
Jefatura	€2.500.000

Tabla de Pago (Servicios profesionales)	
Puesto	Salario
Consultor	€10.000 por hora

Ejemplos:

- Si Juan es **programador** y está nombrado en una jornada de **48 horas**, tiene un salario base de **€1.800.000** (*lo que indica la tabla de la izquierda*).
- Si Beto es **programador** y está nombrado en una jornada de **24 horas**, tiene un salario base de **€900.000** (*la mitad del monto indicado en la tabla de la izquierda*).
- Si Luis tiene una **Jefatura** y está nombrado en una jornada de **12 horas**, tiene un salario base de **€625.000** (*la cuarta parte del monto que indica la tabla de la izquierda*).
- Si Pedro es **consultor** y trabaja **5 horas**, tiene un salario de **€50.000** (*según la tabla de la derecha*)

2.2. Función calcularBono (8 pts.)

Esta función debe recibir por parámetro los **años de experiencia**, el **puesto** y el **salario base** de un trabajador. La función debe calcular y retornar el monto del bono por concepto de años de experiencia del trabajador, este es un monto adicional que recibirá el trabajador.

Este bono se calcula con base en las siguientes reglas:

- Si el trabajador tiene entre 5 y 10 años de experiencia, recibirá un bono adicional del 5% sobre el **salario base**.
- Si el trabajador tiene entre 11 y 20 años de experiencia, recibirá un bono adicional del 15% sobre el **salario base**.
- Si el trabajador tiene más de 20 años de experiencia, recibirá un bono adicional del 20% sobre el **salario base**.

Estas reglas aplican para todos los puestos, a excepción de los consultores, quienes no reciben este bono, pues no son empleados de la empresa registrados en planilla.

2.3. Función calcularSalarioNeto (8 pts.)

La función debe recibir por parámetro los montos del **salario base** y del **bono** de un trabajador. La función debe calcular y retornar el **salario neto** del trabajador. Este monto corresponde a la sumatoria del **salario base** del trabajador más el **bono** (si lo hubiera), menos las siguientes deducciones de ley, las cuales corresponden a un porcentaje que se aplica sobre el **salario base** del trabajador:

- CCSS:
 - 9%
- Renta:
 - 5% en montos entre 600.000 y 1.000.000
 - 15% en montos mayores a 1.000.000 y menores o iguales a 2.000.000
 - 20% en montos mayores a los 2.000.000

Estas deducciones aplican para todos los puestos, excepto para los consultores.

2.4. Función main() (11 pts.)

El main deberá permitir al usuario ingresar los datos y mostrar la información salarial de n trabajadores (n lo determina el usuario). Para cada trabajador se debe ingresar: **nombre**, **puesto**, **años de experiencia** y **horas laboradas**. Para cada trabajador se deben mostrar en pantalla los montos del **salario base**, del **bono** y del **salario neto**. Para el cálculo de estos montos se deben utilizar adecuadamente las funciones especificadas en los puntos 2.1, 2.2 y 2.3.

III Parte: Desarrollo (35 pts.)

3.1. Función verificarBinario (15 pts.)

Escriba una función que reciba por parámetro un número entero y verifique si el número está compuesto solamente por dígitos 1 o 0, en cuyo caso el numero entero representa un numero en binario. De ser así, la función deberá retornar **true**, en caso contrario, deberá retornar **false**.

Por ejemplo:

n = 100001	R/ true
n = 110031	R/ false
n = 11	R/ true

3.3. Función andBinario (20 pts.)

Escriba una función que reciba como parámetros dos números enteros, compuestos por dígitos 1 y 0, los cuales representan números en binario, por ejemplo: 11101 y 11111. Los números recibidos pueden tener diferente cantidad de dígitos.

La función deberá realizar la operación lógica **AND** con cada par de dígitos correspondientes de los dos números recibidos. La función debe generar y devolver un tercer número entero con el resultado de la aplicación de la operación AND a los dos números recibidos.

Observe algunos ejemplos:

Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4
$\begin{array}{r} 1 \ 1 \ 1 \ 0 \ 1 \ 0 \\ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \hline R/ \quad 1 \ 1 \ 1 \ 0 \ 1 \ 0 \end{array}$	$\begin{array}{r} 1 \ 1 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \\ \hline R/ \quad 1 \ 0 \ 1 \ 0 \end{array}$	$\begin{array}{r} 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \\ \hline R/ \quad 0 \ 0 \ 1 \ 0 \ 1 \ 0 \end{array}$	$\begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\ \hline R/ \quad 0 \ 0 \ 0 \ 1 \ 0 \ 0 \end{array}$

Nota: Considere que los números en cualquier base tienen dígitos cero (0) no significativos a la izquierda, tal como se muestra en gris en los últimos dos ejemplos.