

El test debe ser contestado en la **hoja de lectura óptica**. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta.

El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 preguntas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio.

Cada respuesta correcta: 1 punto. Respuesta incorrecta o en blanco: 0 puntos.

- Una constante puede pasarse como argumento a una función:
 - Sólo por valor
 - Sólo por referencia
 - Por valor y por referencia
 - No puede pasarse
- Dado el siguiente fragmento de programa en C±:


```
typedef Uno Dos[Tres];
```

 - Uno es un tipo cualquiera y Tres es una variable entera
 - Dos es un tipo cualquiera y Uno es una constante entera
 - Dos es un tipo cualquiera y Tres es una constante entera
 - Uno es un tipo cualquiera y Dos es un tipo vector
- La medida de complejidad O evalúa de un programa:
 - La cantidad de memoria que consume al ejecutarse
 - La cantidad de memoria que consume al compilarse
 - El tiempo que tarda en ejecutarse
 - Lo difícil que es desarrollarlo
- Dada la función `comp` escrita a continuación (la función `strlen` devuelve la longitud de una cadena):


```
typedef char Vchar[40];
int comp(Vchar c1, Vchar c2){
    int i, j;
    i = strlen(c1)-1; j = strlen(c2)-1;
    while(i >= 0 && j >= 0){
        if(c1[i] != c2[j]) {
            return (int(c1[i]) - int(c2[j]));
        }
        i = i-1; j = j-1;
    }
}
```

 Si el argumento `c1` es "MONT" y `c2` es "MONK":
 - La función devuelve un valor mayor que 0
 - La función devuelve 0
 - La función devuelve un valor menor que 0
 - La función devuelve 1
- Para poder escribir un resultado desde un programa:
 - Sólo es necesario utilizar el procedimiento `printf`
 - Es necesario utilizar la directiva `#include` y el procedimiento `printf`
 - Sólo es necesario utilizar la directiva `#include`
 - Es necesario utilizar la directiva `#include` o bien el procedimiento `printf`
- Después de la ejecución del siguiente fragmento de código en C±:


```
n = 1; res = 2;
while(n < 3){
    res = res*res; n++;
}
```

 - `res` toma el valor 4
 - `res` toma el valor 256
 - `res` toma el valor 2
 - `res` toma el valor 16
- Dentro de un `struct`:
 - Pueden declararse variables, funciones y procedimientos
 - Pueden declararse variables, y definirse funciones y procedimientos
 - Pueden declararse variables y funciones, y definirse procedimientos
 - Sólo pueden declararse variables
- Señale cuál de las siguientes afirmaciones es falsa:
 - La interpretación de programas es más lenta que la ejecución de programas compilados
 - Cada tipo de máquina requiere un tipo de programa objeto
 - Un intérprete produce programas objeto para una máquina virtual
 - Un compilador transforma un programa fuente en un programa objeto
- Una característica deseable de los módulos de un programa es que:
 - Su acoplamiento sea mínimo y su cohesión máxima
 - Su acoplamiento sea mínimo y su cohesión mínima
 - Su acoplamiento sea máximo y su cohesión máxima
 - Su acoplamiento sea máximo y su cohesión mínima
- Dado el siguiente fragmento de programa en C±:


```
while (n > 0) { p = p*n; ??; }
```

 El bucle siempre finaliza cuando:
 - El valor de `p` antes de comenzar el bucle sea negativo y la sentencia `??` sea `p++`
 - El valor de `n` antes de comenzar el bucle sea positivo y la sentencia `??` sea `n--`
 - El valor de `n` antes de comenzar el bucle sea positivo y la sentencia `??` sea `p--`
 - El valor de `p` antes de comenzar el bucle sea negativo y la sentencia `??` sea `n++`

EJERCICIO DE PROGRAMACIÓN

Realizar en C± el TAD **Contribuyentes** para almacenar el nombre y el salario de 50 personas. Dicho tipo contará con las siguientes operaciones. La operación **InicializarSalario** pone a cero el salario de todos los contribuyentes. La operación **OrdenarContribuyentes** ordena a los contribuyentes de mayor a menor salario.