

# Tecnólogo en Informática - Principios de Programación - curso 2007

## Práctico 6: las estructuras de iteración condicional

### Ejercicio 1

Determine el valor que se exhibe en la salida estándar al ejecutar los siguientes fragmentos de código, para variables enteras:

- i) 

```
x = 10;
while (x > 0)
    x = x - 3;
printf("%d", x);
```
- ii) 

```
suma = 0;
i = 3;
while (i <= 7)
{
    suma = suma + i;
    i = i + 2;
}
printf("%d %d", i, suma);
```
- iii) 

```
suma = 0;
scanf("%d", &x);
while (x >= 0)
{
    suma = suma + x;
    scanf("%d", &x);
}
printf("%d", suma);
```

Asuma que los valores leídos de la entrada estándar son 10, 5, 12 y -5.

- iv) 

```
n = 3;
suma = 0;
i = 0;
while (i < n)
{
    scanf("%d", &valor);
    if (valor > 0)
        suma = suma + valor;
    else
        i = i + 1;
}
printf("Fin de la prueba %d %d %d", suma, i, valor);
```

Asuma que los valores leídos de la entrada estándar son 5, 6, -3, -4, 7, 0, 5, 8 y 9

### Ejercicio 2

Determine el valor que se exhibe en la salida estándar al ejecutar los siguientes fragmentos de código, para variables enteras:

- i) 

```
a = 6;
b = 5;
do {
    a = a + 1;
} while (a <= b);
printf("%d", a);
```
- ii) 

```
i = 0;
```

```

do {
    printf("Se imprime.");
} while (i != 0);

```

```

iii) a = 3;
      b = 5;
      do {
          a = a + 1;
          printf("%d", a);
      } while (a <= b);

```

### Ejercicio 3

Determine cuáles de los siguientes segmentos de código producen la misma salida al ser ejecutados:

```

i)    i = 1;
      j = 2;
      do {
          printf("%d %d", i, j);
          i = i+1;
          j = j+1;
      } while (j > 3);

```

```

ii)   i = 1;
      j = 1;
      while (i <= 3 && j <= 2)
      {
          printf("%d %d", i, j + 1);
          i = i + 1;
          j = j + 1;
      }

```

```

iii)  i = 1;
      do {
          printf("%d %d", i, i+1);
          i = i+1;
      } while (i > 3);

```

### Ejercicio 4

Reescriba el código siguiente usando **while** en vez de **do ... while**.

```

do {
    cuerpo
} while (condicion);

```

### Ejercicio 5

Escriba un programa en C que lea de la entrada estándar valores enteros hasta que se ingrese un valor negativo cualquiera y luego despliegue la cantidad de valores no negativos ingresados.

### Ejercicio 6

Implemente en C el algoritmo del ejercicio 6 del práctico 1.

### Ejercicio 7

Se desea tener un programa que calcule el saldo de una cuenta. Suponga que los datos son leídos de la entrada estándar y que constan de renglones, cada uno de los cuales contiene una letra en la primera columna, seguida de un valor real. El último renglón contiene únicamente la letra X en la columna uno. El primer renglón contiene la letra A y el saldo anterior de una cuenta de cheques. Los demás renglones contienen la letra D y el importe de un depósito o la letra R y

el importe de un retiro. Escriba un programa en C que determine el saldo exacto de la cuenta después de procesar las transacciones.

**Ejemplo:**

```
A 1200.35
D 64.12
R 390.00
R 289.67
D 13.02
R 51.07
X
El saldo final es 546.75
```

**Ejercicio 8**

Escribir un programa en C para el algoritmo del ejercicio 7 del práctico 1 y otro para la modificación al mismo algoritmo que se propone en el ejercicio 8 del práctico 1.

**Ejercicio 9**

Escribir un programa en C que determine si un número  $n$  es primo.  $n$  es un entero positivo leído de la entrada estándar.

**Ejercicio 10**

Realizar las funciones de una calculadora simple, teniendo en cuenta que:

- Los datos de entrada serán una secuencia de dígitos decimales y de operadores +, \*, /, -, seguida del signo =.
- Los números y los operadores se ingresan alternados.
- El último elemento ingresado antes del símbolo = es siempre un número.
- Se ignoran los espacios en blanco.
- Los operadores se aplican en el orden en que aparecen en los datos de entrada (no se tiene en cuenta la precedencia de los operadores definida en C), y producen resultados enteros.
- Se asume que se ingresa al menos un número.

```
Ej. de entrada: 4 + 3 / 2 * 8 - 4 =
Ej. de salida: 20
```

**Ejercicio 11**

Escriba un programa en C que reciba como parámetros dos enteros  $b$  y  $n$  y devuelva como resultado la parte entera del logaritmo de  $n$  en base  $b$ . Dicho resultado es un entero  $k$  que cumple lo siguiente:  $b^k \leq n < b^{(k+1)}$  El algoritmo sólo puede realizar divisiones y sumas. Asuma que  $n > 0$  y  $b > 1$ .