Programación 2009

Tarea 2

Las preguntas precedidas por un asterísco son para los alumnos de maestría. En licenciatura, dan puntos extras.

1 Operadores

Ejercicio 1.1 *Evaluar les expresiones (**independientes**) siguientes con los valores **enteras** $A = 20$, $B = 5$, $C = -10$, $D = 2$, $X = 12$ y $Y = 15$. Dar el resultado y especificar cuales son las variables que han cambiado de valor por las instrucciones.*

1. (5*(X+1)/2)*(B+8)
2. A == (B*=4)
3. C != (A /= -D)
4. A %= –D
5. (–X)*(A+C)
6. A = X*(B<C)+Y*!(B<C)
7. A&&B||!0&&C&&!D
8. ((!A&&B)||!0)&&(C&&(!D))

Ejercicio 1.2 *Escribir un programa usando operadores de shifts de bits, que tome en entradas dos enteros de tipo **short** y que les almacene internamente **ambos** en uno y uno solo **unsigned int**. Hacer funciones que permitan recuperar cada de los dos short a partir del **unsigned int**.*

***Ejercicio 1.1** *Escribir una función que tome de entrada un entero de tipo **unsigned long** y que le transforme en otro entero del mismo tipo con los bits movidos de un lugar a la derecha, **circularmente**.*

2 Estructuras de control

Ejercicio 2.1 *Escribir un programa que permita re-arreglar los datos de un arreglo dado, con valores entre 0 y 9, de tal manera que todos los elementos cuyo valor es ≤ 2 se encuentren al principio del arreglo, todos los elementos cuyo valor es entre ≥ 3 y ≤ 6 sean al centro del arreglo, y todos los otros (valores ≥ 7) al final. Un ejemplo sería:*

```
arreglo inicial :
5 8 1 4 3 9 2 7 3 8 1 4 5 3 8

arreglo re-ordenado :
1 2 1 4 3 5 3 4 3 5 8 7 9 8 8
```

Dar una estimación de la complejidad de su algoritmo (no se necesita que sea eficiente, sólo que haga lo pedido).

***Ejercicio 2.1** *Escribir un programa que pida un flotante ϵ al usuario y que calcula una aproximación de e con precisión de ϵ usando la serie de Taylor de la función e^x en $x = 1$.*

3 Arreglos estáticos

Ejercicio 3.1 *Una manera de acelerar programas es reemplazar llamadas de funciones matemáticas de soporte finito por el uso de unas **tablas de valores**. Por ejemplo, si se necesitan frecuentemente los valores de los cuadrados de números entre 0 y 100, entonces mejor guardar todas los valores de esos cuadrados, calculados una vez por todas, en un arreglo de 100 enteros. Basándose en esa idea de tablas de valores, re-escribir lo siguiente para que vaya mas rápido.*

```
#include <stdio.h>
#include <math.h>
int main() {
  int iteration, i;
  const float pion180 = M_PI/180.0;
  for (iteration=0;iteration<1000;iteration++) {
    for (i=0;i<360;i++) {
        float val = fabs(tan(pion180*i));
    }
  }
  return 0;
}
```