
Fundamentos de Programación

Sesión 2

Actividades a realizar en casa

Actividad: Resolución de problemas.

Resolved los siguientes problemas de la relación 1. Recordad que, antes del inicio de esta sesión en el aula de ordenadores, hay que subir las soluciones a PRADO. Se debe subir un fichero cpp por cada ejercicio.

- 5 (Fabricante)
- 7 (Media aritmética y desviación típica)
- 8 (Pinta dígitos)
- 9 (Tiempo)
- 10 (Máquina)

Actividad: Buscar información.

Recordad, o buscar en internet, cómo se pasa un número en base decimal a binario y, al revés, cómo se pasa de binario a decimal.

Actividades de ampliación (voluntarias)

Recordad los conceptos de combinación y permutación, que irán apareciendo recurrentemente a lo largo de la carrera. Consultad, por ejemplo

<http://www.disfrutalasmatematicas.com/combinatoria/combinaciones-permutaciones.html>

Aunque basta con buscar en Google “Permutaciones combinaciones”, “Combinatoria” o “Análisis combinatorio” para encontrar muchas referencias en la web.

Ejecutad el applet

<http://www.disfrutalasmatematicas.com/combinatoria/combinaciones-permutaciones-calculadora.html>

con los valores pertinentes para calcular cuántos número distintos se pueden representar utilizando únicamente dos símbolos (0 y 1) en 64 posiciones de memoria. Ejecutad el mismo applet para que muestre en la parte inferior las distintas permutaciones que se pueden conseguir con 2 símbolos (0,1) y 4 posiciones.

Actividades a realizar en las aulas de ordenadores

El profesor pedirá a algunos alumnos elegidos aleatoriamente que resuelvan en papel los ejercicios indicados en la página anterior. Después se corregirán. Mientras tanto, el resto de alumnos deben intentar resolver los ejercicios siguientes de la relación 1. Estos ejercicios han de entregarse a través de PRADO.

25 (Funciones matemáticas)

26 (Locomotoras)

27 (Triángulos)

Algunas cuestiones a tener en cuenta sobre los tipos de datos numéricos

Los tipos de datos numéricos puede presentar ciertos “errores” al utilizarlos en un programa (no solo en C++). Este tipo de “errores”, que, en principio, un programador inexperto puede achacar a un mal funcionamiento del compilador, son debidos al tipo de dato que el compilador asigna por defecto a los literales y a la forma de representación interna de ellos. Veamos dos ejemplos:

1. Literales de tipo entero. Implementa el siguiente programa y comprueba el resultado.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main () {
6      // Calculo de la media
7      int valor1=3, valor2=4;
8      double media;
9
10     media = (valor1 + valor2) / 2;
11
12     cout << "La media es: " << media;
13 }
```

2. Literales de tipo real. Implementa los siguientes programas y comprueba el resultado.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main () {
6      float f = 1.0/2.0;
7      if (f == 1.0/2.0)
8          cout << "Es lo mismo" << endl;
9      else
10         cout << "No es lo mismo" << endl;
11 }
```

```
1
2 #include <iostream>
3
4 using namespace std;
5
6 int main (){
7
8     float a = 12335545621232154;
9     a = a + 1;
10    a = a - 12335545621232154;
11
12    cout << "a vale: " << a << '\n';
13 }
14
```

¿Qué explicación le das a los resultados que se obtienen?