

# CC1001-7 Auxiliar Extra – Control 1

Daniel Calderón – Sebastián Fehlandt

31 de Marzo de 2009

## P1) Intersectar 2 rectas

Dos rectas, representadas por los puntos  $(X1, Y1) - (X2, Y2)$  y  $(X3, Y3) - (X4, Y4)$ , pueden: 1) Intersectarse en un punto, 2) No Intersectarse, y 3) Intersectarse en infinitos puntos. Se pide escribir un programa que permita determinar en qué punto se intersectan dos rectas, basado en el siguiente diálogo:

Ingrese X1? 1.0	Ingrese X3? 0.0
Ingrese Y1? 0.0	Ingrese Y3? 0.0
Ingrese X2? 1.0	Ingrese X4? -4.0
Ingrese Y2? 5.5	Ingrese Y4? -4.0
Las rectas se intersectan en: (1.0, 1.0)	

## P2) Validador de RUN

El sistema de identificación que se usa en Chile se basa en un código (RUN) que corresponde a un número entero y un dígito verificador (que se calcula usando el número anterior) para saber si algunos de los dígitos del carné han sido modificados. Así, para saber si la cédula 12.575.538-1 está correcta, debemos calcular el dígito verificador para 12575538 y compararlo con 1. Si son iguales, el número está bien, si no, fue adulterado.

Para calcular el dígito verificador de un número se utiliza el siguiente algoritmo. Suponga que la parte numérica del RUN está compuesta por 8 dígitos, representados por los caracteres  $c1$  a  $c8$ . Luego, el dígito verificador se calcula como determinando el resultado de las siguientes operaciones, representadas por la variable  $sum$ :

$$sum = c_8 * 3 + c_7 * 2 + c_6 * 7 + c_5 * 6 + c_4 * 5 + c_3 * 4 + c_2 * 3 + c_1 * 2$$

Luego, el dígito verificador queda definido por la siguiente función:

$$digitoVerificador(sum) = \begin{cases} 0 & \text{si } (sum \% 11) = 0 \\ k & \text{si } (sum \% 11) = 1 \\ 11 - (sum \% 11) & \text{sino} \end{cases}$$

1. Escriba un método que retorne el dígito verificador (10 si es k) asociado cuyo encabezado sea:  
**public static int digitoVerificador(int run).**
2. Escriba un programa que utilice el método digitoVerificador para calcular el dígito verificador de los RUN ingresados por el usuario, basados en el siguiente diálogo:

```
Calculo de dígito verificador
Para terminar escriba 0

Ingrese su Rut sin digito verificador: 15780984
El digito verificador es 9

Ingrese su Rut sin digito verificador: 9845215
El digito verificador es: K
.
.
.
Ingrese su Rut sin digito verificador: 0
```

## P3) Población de Pingüinos

Se estimaba que la población de pingüinos en el 2006 era de 24530. Los biólogos marinos están preocupados, pues saben que la corriente del niño (que ocurre cada aproximadamente 10 años y la última fue en 1997), mata a la mitad de ellos. Adicionalmente, entre 1000 y 2000 (número aleatorio) pingüinos mueren cada año debido a la contaminación de las aguas y a la caza de los mismos. Por otra parte, los estudios revelan que si se consideran los nacimientos y muertes naturales, la población de pingüinos aumenta en un 50% cada año.

Además, el medio ambiente proporciona alimento sólo para 40000 pingüinos, todos los demás morirán de hambre. Implemente un programa que dado un año entregue una estimación de la población de pingüinos. Para lograr esto se pide:

- a) Implemente la función **public static int nPinguinoR(int year, int lastyear, int lastreg)** que retorne la cantidad de pingüinos para el año ingresado como parámetro, siendo lastyear el último año donde se contabilizaron los pingüinos y lastreg la población de aquel entonces.
- b) Implemente la función **public static void consulta()** que siga el siguiente diálogo:

```
Ingrese año del último registro? 2006
Ingrese población del último registro? 24530
Ingrese año para realizar la estimación? 2020
La población estimada para el año 2020 es de X pingüinos.
```

- c) Implemente la función **public static void tabla()** que siga el siguiente diálogo:

```
Ingrese año del último registro? 2006
Ingrese población del último registro? 24530
Ingrese año de partida de la tabla? 2009
Ingrese año de término de la tabla? 2100

Año      Población      Estado
2009     24500           fuera de peligro
2010     22355           fuera de peligro
...
2100     35000           sobrepoblación
```

Donde el estado es: extinto (si  $X=0$ ), en peligro de extinción (si  $0 < X \leq 10000$ ), fuera de peligro (si  $10000 < X \leq 35000$ ) y sobrepoblación (si  $35000 < X$ ).

- d) Implemente un programa que utilice las funciones anteriores para seguir el siguiente dialogo:

```
Estimador de la población de pingüinos
Seleccione una opción:
1. Consultar un año particular
2. Mostrar una tabla con las poblaciones estimadas
3. Salir
opción?
```

#### P4) (Propuesto) Invertir un número

- a) Escriba un método de encabezado **static public int severla(int x)** que reciba un número entero y lo entregue invertido.
- b) Escriba un programa **static public void main(String [] args)** que le pida a usuario un número entero y lo escriba invertido, un número indeterminado de veces, finalizando con "0". Además el programa debe avisar cuando se trate de un número palíndromo, mediante el siguiente dialogo:

```
Ingrese un número entero: 34
Su número invertido es: 43
Ingrese un número entero: 989
Su número invertido es: 989, y se trata de un palíndromo.
Ingrese un número entero: 0
Fin del programa
```