

Tarea 1

Inteligencia Artificial (CC52A)

Profesor: Carlos Hurtado L.
Auxiliar: Renato Valenzuela P.

Fecha de Entrega: 25 de Octubre de 2007.

Ejercicios basados en problemas aparecidos en alguna versión de la ACM International Collegiate Programming Contest (ICPC).

Problema 1

WERTYU



Un error común al escribir en un computador es colocar las manos en el teclado una columna a la derecha de la posición correcta. Así, “Q” se escribe como “W” y “J” se escribe como “K”, etc. El trabajo de usted es decodificar un mensaje escrito de esta manera.

La función a realizar recibe una línea de texto. Esta línea puede contener dígitos, espacios y letras mayúsculas (excepto Q, A, Z), o puntuaciones mostradas arriba [excepto el acento grave (‘)]. Las teclas marcadas con palabras [*Tab*, *BackSp*, *Control*, etc] no estarán dentro de la línea. Debe reemplazar cada letra o símbolo de puntuación por el que está inmediatamente a su izquierda en el teclado QWERTY mostrado arriba. Los espacios en la línea deben ser mostrados iguales.

Se le recomienda utilizar la función de prolog *name*, que funciona así:

name(‘hola’, X). $\longrightarrow$ X = [104, 111, 108, 97]

Es decir, pasa de un string a su representación como lista de caracteres.

Sample Input

O S, GOMR YPFSU/

Output for Sample Input

I AM FINE TODAY.

En Prolog:

p1(‘O S, GOMR YPFSU/’, Resp).
 $\Rightarrow$ Resp = ‘I AM FINE TODAY’

Problema 2

The $3n + 1$ problem.

Considere el siguiente algoritmo:

```
input n
while true
  print n
  if  $n == 1$ 
    break
  else
    if  $n$  es impar
       $n \leftarrow 3n + 1$ 
    else
       $n \leftarrow n/2$ 
```

Si de entrada utilizamos el número 22, se imprimirá la siguiente secuencia de números: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.

Está conjeturado que el algoritmo descrito efectivamente termina (n llega a 1) para cualquier input entero. A pesar de la simplicidad del algoritmo, no se sabe aún si esta conjetura es verdad. Sin embargo, se ha demostrado que sí para todo entero n tal que $0 < n < 1000000$ (y, de hecho, para muchos más números que esto.)

Para una entrada n , es posible determinar la cantidad de números que serán mostrados en pantalla (incluido el 1). Para un número n , esto se llama el *largo del ciclo* de n . En el ejemplo anterior, el largo del ciclo de 22 es 16. Para dos números cualquiera i y j , usted debe determinar el máximo largo de ciclo sobre todos los números entre i y j .

La entrada a su función serán dos enteros, i y j , ambos menores que 1000000 y mayores que 0. Con esto, la función debe retornar el máximo largo de ciclo entre esos dos números.

Sample Input

```
1 10
100 200
201 210
900 1000
```

Sample Output

```
20
125
89
174
```

En Prolog:

```
p2(1, 10, Resp).
=> Resp = 20
```

```
p2(100, 200, Resp).
=> Resp = 125
...
```

Problema 3

Marbles in Three Baskets

Cada una de tres canastas contiene un cierto número de canicas. Se puede mover de una canasta hacia otra tantas canicas como hay actualmente en esa última canasta, y por lo tanto, duplicando la cantidad en la canasta que recibió las canicas. Usted debe encontrar una secuencia de movimientos que terminen con el mismo número de canicas en las tres canastas. Aún más, usted debe lograr dicho cometido en el menor número posible de movimientos. Su programa también debe reconocer el caso en donde no exista tal secuencia.

Sin embargo, para simplificar un poco el problema y para que se cumpla el objetivo de la tarea de que programen en prolog, simplemente, dado una configuración inicial de canicas en las tres canastas, usted debe responder si es que se puede o no se puede llegar a una configuración final de las tres canastas con el mismo número de canicas siguiendo la regla mencionada anteriormente.

Cuidado con ciclar infinitamente revisando las mismas configuraciones una y otra vez.

Sample Input

```
6 7 11
15 18 3
5 6 7
```

Sample Output

```
Yes
Yes
No
```

En Prolog:

```
p3(6, 7, 11).
=> Yes
```

```
p3(15, 18, 3).
=> Yes
```

```
p3(5, 6, 7).
=> No
```