

Tipos Primitivos de Dados

Problemas resolvidos

Prof. Edson Alves

2018

Faculdade UnB Gama

1. AtCoder Beginner Contest 106 – Problem C: To Infinity
2. OJ 11879 – Multiple of 17
3. URI 1163 – Angry Ducks

AtCoder Beginner Contest 106 – Problem C: To Infinity

Problema

Mr. Infinity has a string S consisting of digits from 1 to 9. Each time the date changes, this string changes as follows:

- Each occurrence of '2' in S is replaced with '22'. Similarly, each '3' becomes '333', '4' becomes '4444', '5' becomes '55555', '6' becomes '666666', '7' becomes '7777777', '8' becomes '88888888' and '9' becomes '999999999'. '1' remains as '1'.

For example, if S is '1324', it becomes '1333224444' the next day, and it becomes '13333333332222444444444444444444' the day after next. You are interested in what the string looks like after 5×10^{15} days. What is the K -th character from the left in the string after 5×10^{15} days?

Constraints

- $2 \leq K \leq N \leq 100000$
- A_1, A_2, \dots, A_N is a permutation of $1, 2, \dots, N$

Input

Input is given from Standard Input in the following format:

$$\begin{array}{c} N \quad K \\ A_1 \quad A_2 \quad \dots \quad A_N \end{array}$$

Output

Print the K -th character from the left in Mr. Infinity's string after 5×10^{15} days.

Exemplo de entradas e saídas

Exemplo de Entrada

1214

4

3

157

299792458

9460730472580800

Exemplo de Saída

2

3

2

- O primeiro fato a ser observado é, se o número inicial x da string for diferente de 1, ele será replicado $x^{5 \times 10^{15}}$ vezes, de modo que a resposta será o próprio x
- O caso especial ocorre quando a string é prefixada por uma sequência de 1s
- Se a quantidade de uns for maior ou igual a K , a resposta será igual a 1
- Caso contrário, a resposta será igual ao primeiro caractere da string diferente de 1

```
1 #include <iostream>
2
3 using namespace std;
4 using ll = long long;
5
6 char solve(const string& S, ll K)
7 {
8     for (const auto& c : S)
9     {
10         if (c != '1')
11             return c;
12
13         if (not (--K))
14             break;
15     }
16
17     return '1';
18 }
19
```



```
20 int main()
21 {
22     ios::sync_with_stdio(false);
23
24     string S;
25     ll K;
26
27     cin >> S >> K;
28
29     cout << solve(S, K) << '\n';
30
31     return 0;
32 }
```

OJ 11879 – Multiple of 17

Theorem: *If you drop the last digit d of an integer n ($n \geq 10$), subtract $5d$ from the remaining integer, then the difference is a multiple of 17 if and only if n is a multiple of 17.*

For example, 34 is a multiple of 17, because $3-20=-17$ is a multiple of 17; 201 is not a multiple of 17, because $20-5=15$ is not a multiple of 17.

Given a positive integer n , your task is to determine whether it is a multiple of 17.

Input

There will be at most 10 test cases, each containing a single line with an integer n ($1 \leq n \leq 10^{100}$).

The input terminates with $n = 0$, which should not be processed.

Output

For each case, print 1 if the corresponding integer is a multiple of 17, print 0 otherwise.

Exemplo de entradas e saídas

Exemplo de Entrada

34

201

2098765413

1718

0

Exemplo de Saída

1

0

1

0

- O limite máximo da entrada não pode ser armazenada em tipos primitivos do C/C++
- Uma alternativa é ler a entrada como strings, e aplicar o algoritmo descrito, acumulando no total S , para cada dígito d , o valor $-5d$
- Em seguida, basta verificar se S é ou não múltiplo de 17
- Outra alternativa é utilizar uma linguagem que tenha suporte nativo para aritmética estendida
- Uma opção é a linguagem Java e sua classe BigInteger
- Outra opção é usar a linguagem Python

```
1 import sys
2
3 xs = [int(x) for x in sys.stdin.readlines()][:-1]
4 ans = ['1' if x % 17 == 0 else '0' for x in xs]
5
6 print('\n'.join(ans))
```

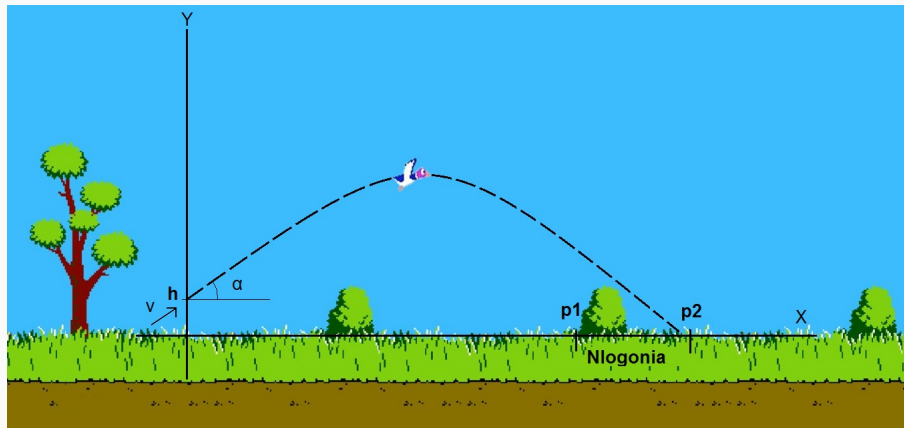
URI 1163 – Angry Ducks

Problema

Em uma terra distante existem duas cidades, a Nlogônia onde vivem os Nlogoneses, e Ducklogônia onde vivem seus vizinhos os Duckneses, já há algum tempo estas duas cidades estão em guerra e agora em uma tentativa de ganhar a guerra os Duckneses pretendem atacar a cidade da Nlogônia com um bodoque que atira patos, porem para que não haja erro eles pediram que você construa um programa que dados os valores da altura do bodoque (h), os pontos onde inicia ($p1$) e onde termina ($p2$) a cidade da Nlogônia, o ângulo do disparo (α) e a velocidade do lançamento, calcule se o projétil atingira o alvo.

Para os cálculos assuma que a aceleração da gravidade é $g = 9.80665$ e que $\pi = 3.14159$.

Problema



Entrada

Existem vários casos de teste, cada caso inicia com 1 valor de ponto flutuante h ($1 \leq h \leq 150$) indicando a altura do bodoque, a próxima linha contem 2 valores inteiros $p1$ e $p2$ ($1 \leq p1, p2 \leq 9999$) indicando onde inicia e onde termina a Nlogônia, a linha seguinte contem um inteiro n ($1 \leq n \leq 100$) indicando o numero de tentativas que serão feitas para acertar a Nlogônia, as n linhas seguintes contem dois valores de ponto flutuante com os valores do ângulo α ($1 \leq \alpha \leq 180$) e a velocidade V ($1 \leq V \leq 150$) do disparo.

O final do arquivo de entrada é determinado por EOF.

Saída

Para cada disparo, seu programa deve imprimir uma única linha no seguinte formato, “ $X \rightarrow \text{DUCK}$ ” para quando o pato acertar a Nlogônia ou “ $X \rightarrow \text{NUCK}$ ” quando o pato não acertar a Nlogônia, onde X é a distância máxima que o projétil atingiu até chegar ao chão ($Y = 0$). X deve ser impresso com 5 casas decimais.

Exemplo de entradas e saídas

Exemplo de Entrada

2.1

368 380

3

42.3 60

30 55

44 60.876842

Exemplo de Saída

367.76208 -> NUCK

270.72675 -> NUCK

379.83781 -> DUCK

Solução

- Embora não dito explicitamente no texto, a altura y do pato segue uma trajetória quadrática
- De acordo com o movimento uniformemente variado, a altura y do pato no instante t é igual a

$$y(t) = h + v_y t - \frac{g}{2} t^2,$$

onde h é a altura inicial e v_y é a componente y da vetor velocidade, isto é,

$$v_y = v \sin\left(\frac{\alpha\pi}{180}\right)$$

- Já o movimento em x é linear
- De acordo com o movimento uniforme,

$$x(t) = v_x t,$$

onde v_x é a componente x do vetor velocidade, isto é,

$$v_x = v \cos\left(\frac{\alpha\pi}{180}\right)$$

- Assim, basta usar a fórmula de Báskara para determinar o instante \hat{t} onde a altura $y(t)$ é igual a zero:

$$\hat{t} = \frac{-v_y \pm \sqrt{v_y^2 + 2hg}}{2}$$

- Observe que, dentre as duas soluções possíveis, deve ser escolhida a positiva
- A distância percorrida será $x(\hat{t})$ metros
- O alvo será atingido se $p1 \leq x(\hat{t}) \leq p2$

Solução AC

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 const double PI { 3.14159 };
7 const double g { 9.80665 };
8
9 double zero(double h, double v)
10 {
11     double c = h;
12     double b = v;
13     double a = -g/2;
14
15     double delta = b*b - 4*a*c;
16
17     double t1 = (-b + sqrt(delta))/(2*a);
18     double t2 = (-b - sqrt(delta))/(2*a);
19
20     return t1 > 0 ? t1 : t2;
21 }
```



```
22
23 double dist(double h, double vy, double vx)
24 {
25     double t = zero(h, vy);
26     return vx*t;
27 }
28
29 int main()
30 {
31     double h, alpha, v;
32     int p1, p2, n;
33
34     while (cin >> h >> p1 >> p2 >> n)
35     {
36         while (n--)
37         {
38             cin >> alpha >> v;
39
40             double vy = sin(alpha*PI/180.0)*v;
41             double vx = cos(alpha*PI/180.0)*v;
42             double d = dist(h, vy, vx);
```

```
43
44     printf("%.5f -> ", d);
45
46     if (d >= p1 and d <= p2)
47         printf("DUCK\n");
48     else
49         printf("NUCK\n");
50 }
51 }
52
53 return 0;
54 }
```

1. [AtCoder Beginner Contest 103 – Problema C: To Infinity](#)
2. [OJ 11879 – Multiple of 17](#)
3. [URI 1163 – Angry Ducks](#)