

II BXComp

2º Campeonato de Programação para Calouros do Curso de Sistemas de Informação 2012

4ª Etapa – Desafio 2

Triângulo de Pascal da Morte

Sr. Roncato era um pesquisador matemático renomado, especializado em Triângulos de Pascal e suas propriedades. Tinha o hobby de fazer programas de computador que resolvessem problemas matemáticos. Depois que se aposentou, resolveu ensinar sobre Triângulos de Pascal para crianças. Em suas aulas, Sr. Roncato também ensinava programação, além dos principais conceitos de triângulos de Pascal:

- Formação do triângulo de Pascal: cada elemento no triângulo representa um número binomial $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, onde n é o número da linha e k é o número da coluna.

Exemplo:

	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

$$\binom{3}{2} = \frac{3!}{2!(3-2)!} = \frac{3!}{2!.1!} = \frac{3.2.1}{2.1.1} = 3.$$

(linha 3-coluna 2)

- Relação de Stifel: qualquer elemento no interior do triângulo é igual à soma de dois elementos, conforme definição formal: $\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$. Além disso, a primeira linha e as laterais do triângulo são compostas por “1”.

Exemplo:

	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

$$\binom{5}{2} = \binom{5-1}{2-1} + \binom{5-1}{2} = \binom{4}{1} + \binom{4}{2} = 4 + 6 = 10.$$

(linha 5-coluna 2) = (linha4-coluna1) + (linha4-coluna2)

Os alunos gostavam muito da Relação de Stifel. Por isso, Sr. Roncato resolveu que iria desenvolver um algoritmo que gerasse um triângulo de Pascal utilizando a relação de Stifel. Como o triângulo de Pascal é um triângulo numérico infinito, Sr. Roncato precisaria passar um valor limite para a criação desse triângulo. Com esse algoritmo, o Sr. Roncato poderia ensinar às crianças sobre triângulos de Pascal mais facilmente. Além disso, permitiria que ele aprofundasse os conceitos de programação já conhecido por seus alunos. Infelizmente, por problemas de saúde, o Sr. Roncato veio a falecer antes de terminar seu algoritmo.

Tarefa

As crianças, utilizando os conceitos de programação e de triângulos de Pascal aprendidos até o momento, resolveram desenvolver o algoritmo de seu professor. Seu grupo fará o papel dessas crianças e realizará a tarefa.

Dado um valor para um limite (*max*), inteiro e positivo, gerar um triângulo de Pascal (pela Relação de Stifel), de modo que, quando qualquer número no triângulo atingir (ou ultrapassar) o valor de *max*, deve-se finalizar a base do triângulo com a linha que contém esse número. Veja exemplo abaixo.

Max: 10

Triângulo correspondente:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

Entrada

Seu programa deve ler uma série de números positivos e inteiros menores ou iguais a 106, os quais deverão ser usados como valores para limites (*max*). O valor zero (0), no final da série de números da entrada, deverá ser usado como condição de parada de seu algoritmo.

Saída

Para cada teste de entrada (valor para *max*), seu programa deve gerar uma saída da seguinte maneira: uma linha no formato “Max=<teste_entrada>”; *n* linhas com o triângulo de Pascal correspondente (como explicado acima) e uma última linha no formato “Esse triângulo de Pascal tem: <n> linhas.”. Além disso, ao final de todos os casos de testes deve ser impresso “FIM”.

Entre os valores do triângulo há SOMENTE um (1) espaço. NÃO há espaços antes ou ao final de cada linha. Ao final de cada saída é necessário pular uma linha simples. A base do triângulo deve aparecer por completo (dica: não parar no valor de *max*).

Exemplo de Entrada

```
15
32
0
```

Exemplo de Saída

```
Max=15
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
Esse triângulo de Pascal tem: 7 linhas.

Max=32
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
```

Esse triângulo de Pascal tem: 8 linhas.

FIM

(esta saída corresponde ao exemplo de entrada acima)

Restrições

O valor de *max* pode variar da seguinte maneira: $1 \leq max \leq 10^6$.