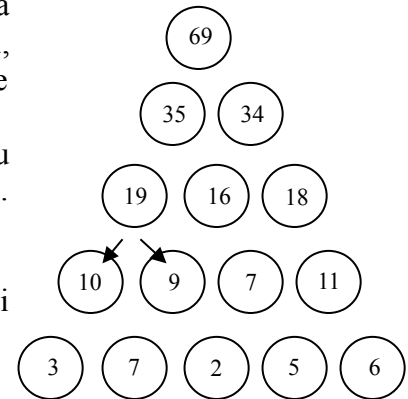


Problema 2 - triunghi

100 puncte

Se consideră un *triunghi* alcătuit din numere naturale scrise pe n linii ca în figura alăturată. Liniile triunghiului sunt numerotate de la 1 la n , începând cu linia de la baza triunghiului (linia de jos), iar pozițiile pe linie sunt numerotate începând cu 1 de la stânga la dreapta.

Fiecare număr din triunghi, exceptând pe cele de pe linia 1, este egal cu suma numerelor aflate imediat sub el, în stânga și respectiv în dreapta lui.



Cerință

Cunoscând câte un număr de pe fiecare linie a triunghiului, determinați toate numerele de pe linia 1.

Date de intrare

Fișierul de intrare `triunghi.in` conține pe prima linie numărul natural n reprezentând numărul de linii din triunghi. Pe următoarele n linii sunt descrise informațiile despre triunghi. Mai exact, pe linia i ($1 \leq i \leq n$) dintre cele n se află două numere naturale separate prin spațiu p_i v_i indicând poziția și respectiv valoarea numărului cunoscut de pe linia i a triunghiului.

Date de ieșire

Fișierul de ieșire `triunghi.out` va conține o singură linie, pe care se găsesc n numere naturale separate prin câte un spațiu, reprezentând în ordinea de la stânga la dreapta numerele scrise pe linia 1 a triunghiului.

Restricții

- $1 \leq n \leq 1000$
- $1 \leq p_i \leq n+1-i$, pentru $1 \leq i \leq n$
- Toate numerele din triunghi sunt numere naturale cu cel mult 18 cifre.
- Pentru elevii care implementează în C: **când utilizați MINGW**, dacă citiți/afișați valori de tip `long long int` folosind `scanf()`, respectiv `printf()` utilizați specificatorul de format `%I64d`.

Exemplu

<code>triunghi.in</code>	<code>triunghi.out</code>	Explicații
5 4 4 2 5 3 13 2 25 1 45	1 2 3 4 2	Triunghiul este: 45 20 25 8 12 13 3 5 7 6 1 2 3 4 2

Timp maxim de execuție/test: 0.2 secunde

Memorie totală: 20MB din care 10MB pentru stivă.

Dimensiunea maximă a sursei 5KB.