

Problema 2018.1.8 - Biblioteca

Cărțile din biblioteca UPB trebuie mutate într-o nouă locație, iar aranjarea lor pe rafturi se va face într-o manieră inovativă, urmărindu-se minimizarea numărului de rafturi necesare. Bibliotecara știe că are un număr suficient de rafturi încât să pună toate cărțile. Pe fiecare raft încap cărți însumând în total D pagini. De asemenea, știe câte cărți cu un anumit număr de pagini există în bibliotecă. Dat fiind acestea și urmărind să ocupe cât mai puține rafturi, bibliotecara și-a propus să aranjeze cărțile pe rafturi:

- i. raft după raft,
- ii. alegând întotdeauna să completeze raftul curent cu cea mai groasă carte disponibilă,
- iii. trecând la următorul raft numai în condițiile în care pe raftul curent nu mai poate fi plasată nicio carte dintre cele rămase și
- iv. asigurându-se că a plasat pe rafturi toate cărțile.

Cerință

Scrieți un program care o poate ajuta pe bibliotecară să aranjeze cărțile pe rafturi în mod eficient, conform regulilor enunțate mai sus.

Date de intrare

Se vor citi de la tastatură (fluxul *stdin*) următoarele date:

- de pe prima linie: două numere întregi D și k , separate prin spațiu, reprezentând D – dimensiunea rafturilor exprimată în număr de pagini, k – numărul de dimensiuni diferite pentru cărțile ce trebuie aranjate în bibliotecă;
- de pe următoarele k linii: câte două numere întregi n și p , reprezentând numărul de cărți n de grosime p pagini ce trebuie aranjate în bibliotecă.

Cele k linii ce conțin informații despre cărți sunt date în ordinea inversă a grosimii p .

Toate liniile conținând date de intrare sunt finalizate cu caracterul *newline* (tasta *Enter*).

Date de ieșire

Programul va afișa pe ecran (*stream*-ul standard de ieșire) m linii, corespunzătoare celor m rafturi pe care a fost plasată cel puțin o carte, în ordinea completării lor (conform regulilor). Fiecare dintre cele m linii va conține o serie de numere întregi, separate prin spațiu, reprezentând dimensiunile cărților ce au fost plasate pe acel raft, în ordinea plasării lor pe raft (conform regulilor).

ATENȚIE la respectarea cerinței problemei: afișarea rezultatelor trebuie făcută EXACT în modul în care a fost indicat! Cu alte cuvinte, pe stream-ul standard de ieșire nu se va afișa nimic în plus față de cerința problemei; ca urmare a evaluării automate, orice caracter suplimentar afișat, sau o afișare diferită de cea indicată, duc la un rezultat eronat și prin urmare la obținerea calificativului „Respins”.

Restricții și precizări

1. Dimensiunea rafturilor D este număr întreg în intervalul $[50; 10000]$.
2. Grosimile cărților p sunt numere întregi în intervalul $[1; 1000]$.
3. Numerele n de cărți de diverse grosimi sunt numere întregi în intervalul $[1; 100]$.
4. Se garantează faptul că nu vor exista cărți de grosime p mai mare decât dimensiunea D a rafturilor.
5. Nu este necesar ca rafturile să fie umplute la dimensiunea maximă D .
6. **Atenție:** În funcție de limbajul de programare ales, fișierul ce conține codul trebuie să aibă

una din extensiile .c, .cpp, .java, sau .m. Editorul web **nu va adăuga automat** aceste extensii și lipsa lor duce la imposibilitatea de compilare a programului!

7. **Atenție:** Fișierul sursă trebuie numit de candidat sub forma: <nume>.<ext> unde nume este numele de familie al candidatului și extensia este cea aleasă conform punctului anterior. Atenție la restricțiile impuse de limbajul Java legate de numele clasei și numele fișierului!

Exemplu

Intrare	Ieșire
200 5 2 130 4 120 2 80 3 60 7 50	130 60 130 60 120 80 120 80 120 60 120 50 50 50 50 50 50 50
Explicație Au fost completate 8 rafturi astfel: Raftul #1: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (130 pag), - apoi cea mai groasă carte disponibilă (60 pag) care mai încapă pe acest raft ($200 - 130 = 70$ pag), - alte cărți nu mai încap în spațiul rămas: $70 - 60 = 10$ pagini. Raftul #2: identic cu raftul #1. Raftul #3: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (120 pag), - apoi cea mai groasă carte disponibilă (80 pag) care mai încapă pe acest raft ($200 - 120 = 80$ pag), - alte cărți nu mai încap pe raft – acesta este completat în întregime. Raftul #4: identic cu raftul #3. Raftul #5: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (120 pag), - apoi cea mai groasă carte disponibilă (60 pag) care mai încapă pe acest raft ($200 - 120 = 80$ pag), - alte cărți nu mai încap în spațiul rămas: $80 - 60 = 20$ pagini. Raftul #6: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (120 pag), - apoi cea mai groasă carte disponibilă (50 pag) care mai încapă pe acest raft ($200 - 120 = 80$ pag), - alte cărți nu mai încap în spațiul rămas: $80 - 50 = 30$ pagini. Raftul #7: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (50 pag), - apoi, în ordine, cele mai groase cărți disponibile care mai încapă pe acest raft: 50, 50, 50, - alte cărți nu mai încap pe raft – acesta este completat în întregime. Raftul #8: <ul style="list-style-type: none"> - cea mai groasă carte disponibilă (50 pag), - apoi, în ordine, cele mai groase cărți disponibile care mai încapă pe acest raft: 50, - au fost epuizate toate cărțile. 	

Timp de lucru: 120 de minute