

1. PREMII // ПРЕМИИ

Fondul de premii pentru concursul ”Aspirații informatice 2019” îl constituie suma de S lei. La concurs participă elevii din clasele IX – XII. În general sunt acordate p_1 premii pentru locul I, p_2 premii pentru locul II și p_3 premii pentru locul III. Se cunoaște că valoarea premiului acordat locului I este de 2 ori mai mare decât cel al premiului pentru locul II. Respectiv, valoarea premiului acordat locului II este de 2 ori mai mare decât cel al premiului pentru locul III.

// Призовой фонд конкурса «Aspirații informatice 2019» составляет сумму S лей. В конкурсе принимают участие ученики IX – XII классов. В общем, присуждаются p_1 премий за I место, p_2 премий за II место и p_3 премий за III место. Известно, что размер премии за I место в 2 раза больше, чем за II место. Соответственно, размер премии за II место в 2 раза больше, чем за III место.

Sarcină. Elaborați un program, care calculează valoarea premiilor acordate pentru fiecare din locuri I-III. // **Задание.** Разработайте программу, которая определяет размер премий, присужденные за каждое из I-III мест.

Date de intrare. Intrarea standard va conține 4 linii: pe prima linie un număr natural S – suma de bani; pe următoarele trei linii numărul de premii acordate pentru fiecare loc p_i , $i = 1, 2, 3$.

// **Входные данные.** Стандартный ввод содержит 4 строки в следующем порядке: в первой строке натуральное число S – призовой фонд в леях; в следующих трех строках количество премий за каждое призовое место p_i , $i = 1, 2, 3$.

Date de ieșire. Ieșirea standard va conține trei linii cu numere reale, reprezentând în ordine: valoarea premiului pentru locul I; valoarea premiului pentru locul II; valoarea premiului pentru locul III.

// **Выходные данные.** Стандартный вывод должен содержать три строки, каждая содержащая одно действительное число, которые обозначает в следующем порядке: размер премии за I место; размер премии за II место; размер премии за III место.

Restricții/Ograničenja: $S \leq 10000$; $0 \leq p_1 \leq 4$; $0 \leq p_2 \leq 8$; $0 \leq p_3 \leq 12$; $S, p_1, p_2, p_3 \in \mathbb{N}$;

- Valoarea premiului este un număr real pozitiv, care are exact două cifre după punctul zecimal; // Размер премии представляет собой положительное действительное число, которое имеет ровно две цифры после запятой;
- Timpul de execuție nu va depăși 1 sec; // Время выполнения программы до 1 сек;
- Programul va folosi cel mult 32 MB de memorie // Максимальный объем памяти – 32 MB.

Exemple // Примеры

INTRARE	IEȘIRE	INTRARE	IEȘIRE	INTRARE	IEȘIRE
1100	400.00	0	0.00	2100	1050.00
1	200.00	1	0.00	1	525.00
2	100.00	2	0.00	2	0.00
3		3		0	

Fișierul sursă va avea denumirea xxx1.PAS, xxx1.C sau xxx1.CPP, unde xxx fiind codul elevului. De exemplu, N011.PAS // Имя исходного файла будет xxx1.PAS, xxx1.C или xxx1.CPP, где xxx – регистрационный код ученика. Например, N011.PAS.

2. PINOCCHIO // ПИНОККИО

În fiecare zi lucrătoare din săptămână, Pinocchio spune câte o minciună datorită căreia nasul acestuia crește cu câte p centimetri pe zi. Sâmbăta și duminica, când vine bunicul Geppetto acasă, pentru a nu-l supăra prea tare, Pinocchio reușește să nu spună nici o minciună, ba chiar uitându-se în oglindă observă că în fiecare din aceste zile lungimea nasului său scade cu câte 1 centimetru pe zi. Când începe o nouă săptămână, rămânând singur acasă, Pinocchio continuă șirul minciunilor.

// В каждый рабочий день недели Пиноккио говорит ложь, которая заставляет его нос расти на p сантиметров в день. В субботу и воскресенье, когда дедушка Джепетто приходит домой, чтобы не огорчать его, Пиноккио умудряется говорить правду и глядя в зеркало, он замечает, что в эти дни длина его носа уменьшается на 1 сантиметр. С начала новой недели Пиноккио, оставаясь дома один, опять продолжает говорить ложь.

Sarcină. Elaborați un program, care determină lungimea nasului lui Pinocchio după k zile, începând de luni, și știind că inițial nasul său măsoară n centimetri. // **Задание.** Разработайте программу, которая определяет длину носа Пиноккио через k дней, начиная с понедельника и зная, что его нос изначально имел длину n сантиметров.

Date de intrare. Intrarea standard va conține trei numere naturale (în această ordine) fiecare din linie nouă: n – lungimea inițială a nasului; p – numărul de centimetri cu care crește nasul în fiecare zi și k – numărul de zile.

// **Входные данные.** Стандартный ввод содержит три натуральных числа в следующем порядке: n – начальная длина носа; p – количество сантиметров, на которое нос увеличивается с каждым днем и k – количество дней.

Date de ieșire. Ieșirea standard va conține un număr natural – lungimea nasului în centimetri după k zile. // **Выходные данные.** Стандартный вывод должен содержать одно натуральное число – длина носа в сантиметрах через k дней.

Restricții // Ограничения:

- $1 \leq n \leq 1000$, $n \in \mathbb{N}$; $1 \leq k \leq 356$, $k \in \mathbb{N}$; $1 \leq p \leq 100$, $p \in \mathbb{N}$;
- Timpul de execuție nu va depăși 1 sec // Время выполнения программы до 1 сек;
- Programul va folosi cel mult 32 MB de memorie // Максимальный объем памяти – 32 MB.

Exemplu // Пример

INTRARE	IEȘIRE
2 1 8	6

Fișierul sursă va avea denumirea xxx2.PAS, xxx2.C sau xxx2.CPP, unde xxx fiind codul elevului, de exemplu, N012.PAS // Имя исходного файла будет xxx2.PAS, xxx2.C или xxx2.CPP, где xxx – регистрационный код ученика. Например, N012.PAS.

3. AUTOBUZ // АВТОБУС

Ionel este un tip nerăbdător și urăște să aștepte. Din păcate, în fiecare zi trebuie să aștepte în stație, pentru că programul autobuzelor este afișat într-un mod foarte ciudat.

În loc să fie afișat timpul de sosire/plecare, pentru fiecare autobuz este afișată lista duratelor rutelor efectuate de autobuzul respectiv.

Mai exact, prin stația lui Ionel trec n autobuze, pe care le vom numerota de la 1 la n . Fiecare autobuz $_i$ circulă pe m_i rute (în ordinea ruta 1, ruta 2, ..., ruta m_i , apoi după terminarea rutei m_i , autobuzul reia de la ruta 1). Pentru fiecare rută este afișată durata necesară parcurgerii rutei respective (exprimată în minute). Autobuzele încep să circule de la momentul 0 (care corespunde zilei de 1 ianuarie, ora 00:00) și circulă în acest mod o anumită perioadă stabilită de Primărie.

// Иван нетерпеливый парень и не любит ждать. К сожалению, каждый день ему приходится ожидать автобуса на остановке, потому что расписание автобусов составлено странным образом. Вместо того, чтобы показывать время прибытия / отправления, для каждого автобуса вывешен список продолжительностей прохождения маршрутов, выполненные соответствующим автобусом.

Более точно, через остановку Ивана проходит n автобусов, которые пронумерованы от 1 до n . Каждый автобус i курсирует по m_i маршрутам (в следующем порядке: маршрут 1, маршрут 2, ..., маршрут m_i , затем после завершения маршрута m_i автобус возобновляет движение по маршруту 1). Для каждого маршрута отображается продолжительность прохождения маршрута (в минутах). Автобусы начинают курсировать с момента 0 (что соответствует дате 1 января, 00:00 часов) и следуют таким образом в течение определенного периода, установленного мэрией.

Sarcină. Elaborați un program, care, cunoscând programul autobuzelor și momentul în care Ionel a venit în stație, să determine câte minute așteaptă Ionel până sosește în stație primul autobuz. // **Задание.** Разработайте программу, которая согласно продолжительностям прохождения маршрутов n автобусами и времени прибытия Ивана на остановку, определит, сколько минут необходимо ждать Ивану прибытия первого автобуса на его остановку.

Date de intrare. Intrarea standard va conține, pe prima linie, un număr natural n , reprezentând numărul de autobuze care circulă prin stația lui Ionel și pe următoarele n linii - rutele celor n autobuze, câte un autobuz pe o linie. Pe linia $i+1$ este un număr natural m_i (numărul de rute parcurse de autobuz $_i$), apoi o succesiune de m_i valori naturale, reprezentând în ordine timpii necesari parcurgerii celor m_i rute. Pe ultima linie este timpul (exprimat în minute) la care sosește Ionel în stație. Numerele de pe aceeași linie sunt separate prin spațiu.

// **Входные данные.** Стандартный ввод содержит в первой строке натуральное число n – количество автобусов, проходящих через остановку Ивана. На следующих n строках содержатся маршруты автобусов, по одному автобусу в строке. В строке $i+1$ содержится натуральное число (количество автобусных маршрутов), затем в той же строке содержится последовательность из m_i натуральных значений, представляющих собой время (в минутах), необходимое для прохождения маршрутов. В последней строке содержится время (в минутах), когда Иван прибывает на остановку. Числа в каждой строке разделены одним пробелом.

Date de ieșire. Ieșirea standard va conține un singur număr natural, reprezentând câte minute trebuie să aștepte Ionel până ce sosește în stație primul autobuz. // **Выходные данные.** Стандартный вывод должен содержать одно натуральное число, представляющее собой время ожидания (в минутах) Иваном прибытия первого автобуса на свою остановку.

Restricții // Ограничения:

- $1 \leq n \leq 20$; $1 \leq m \leq 10$;
- $1 \leq$ timpul necesar pentru parcurgerea unei rute ≤ 100 ; // $1 \leq$ время прохождения маршрута ≤ 100 ;
- $0 \leq$ timpul la care sosește Ionel în stație ≤ 40 ; // $0 \leq$ время прибытия Ивана на остановку ≤ 40 ;
- Dacă timpul la care Ionel sosește în stație coincide cu timpul la care ajunge în stație un autobuz, atunci Ionel prinde autobuzul, deci timpul de așteptare va fi 0; // Если время прибытия Ивана на остановку совпадает со временем прибытия автобуса на эту остановку, то Иван садится в автобус и, в этом случае, время ожидания будет равно 0.
- Timpul de execuție nu va depăși 3 sec // Время выполнения программы до 3 сек;
- Programul va folosi cel mult 32 MB de memorie // Максимальный объем памяти – 32 MB.

Exemple // Примеры

INTRARE	IEȘIRE	INTRARE	IEȘIRE
3	5	3	70
3 40 50 60		9 100 20 30 40 35 25 10 50 60	
3 70 90 30		1 80	
3 100 90 80		2 100 70	
35		10	

Fișierul sursă va avea denumirea xxx3.PAS, xxx3.C sau xxx3.CPP, unde xxx fiind codul elevului, de exemplu, N013.PAS // Имя исходного файла будет xxx3.PAS, xxx3.C или xxx3.CPP, где xxx – регистрационный код ученика. Например, N013.PAS.

4. CARTE // КНИГА

Andrei a primit în dar o carte în care paginile sunt amestecate. Se hotărăște totuși să o citească, răsfoind cartea într-un singur sens, de la prima pagină către ultima, în ordinea așezării lor în carte, respectând următorul algoritm: „Caută la început pagina numerotată cu $x=1$. După ce a citit o pagină cu numărul x caută printre paginile următoare acestei pagini, răsfoind cartea, pagina cu numărul $x+1$, fără a căuta printre paginile așezate înaintea paginii cu numărul x . Dacă o găsește, atunci va continua lectura în același mod, iar dacă nu o găsește atunci va închide cartea și, în ziua următoare, va relua lectura de la pagina cu numărul $x+1$, pe care mai întâi o va căuta răsfoind cartea de la început. Andrei va proceda la fel și în zilele următoare până când va citi întreaga carte”.

// Андрей получил в подарок книгу, в которой страницы перемешаны. Однако он решает прочесть ее, пролистывая страницы книги в одном направлении, от первой страницы до последней, в порядке их размещения в книге, соблюдая следующий алгоритм: «Вначале ищет страницу, пронумерованную с $x=1$. После прочтения одной страницы с номером x , ищет среди следующих страниц от данной страницы, пролистывая книгу, страницу с номером $x+1$, не пролистывая страницы, которые расположены перед страницей x . Если он найдет такую страницу, то продолжит чтение аналогичным образом. В случае если он не найдет такую страницу, то закроет книгу, и на следующий день возобновит чтение со страницы с номером $x+1$, которую он будет искать, пролистывая с самого начала книги. Андрей будет поступать аналогичным образом в ближайшие несколько дней, пока не прочтает всю книгу».

Sarcină. Elaborați un program, care determină numărul zilelor în care Andrei citește cartea, prima zi în care Andrei a citit cele mai multe pagini și numărul paginilor citite în acea zi.

// **Задание.** Разработайте программу, которая определяет количество дней, за которое Андрей прочитает книгу и первый день, в течение которого Андрей прочитал наибольшее количество страниц, и количество страниц, прочитанных в этот день.

Date de intrare. Intrarea standard va conține două linii: pe prima linie un număr natural n – numărul paginilor din carte; pe a doua linie – n numere naturale distincte x_1, x_2, \dots, x_n , separate prin spațiu, reprezentând ordinea în care sunt așezate paginile în carte.

// **Входные данные.** Стандартный ввод содержит две строки в следующем порядке: в первой строке натуральное число n – количество страниц в книге; во второй строке – n натуральных чисел x_1, x_2, \dots, x_n , разделенных одним пробелом, которые обозначают порядок размещения страниц в книге.

Date de ieșire. Ieșirea standard va conține trei numere naturale într-o linie, separate prin spațiu, în următoarea ordine: numărul zilelor în care Andrei citește cartea; numărul primei zile în care Andrei a citit cele mai multe pagini; numărul maxim de pagini citite într-o zi.

// **Выходные данные.** Стандартный вывод должен содержать три натуральных числа, в одной строке, разделенных одним пробелом, которые обозначают в следующем порядке: количество дней, за которое Андрей прочитает книгу; номер первого дня, в который Андрей прочитал наибольшее количество страниц; максимальное количество страниц, прочитанное за этот день.

Restricții // Ограничения:

- $0 \leq n \leq 10001, n \in \mathbb{N}$ (paginile cărții sunt numerotate cu numere naturale distincte de la 1 la n) // (страницы книги пронумерованы различными натуральными числами от 1 до n);
- Citirea cărții presupune citirea fiecărei pagini din carte, o singură dată // Прочтение книги предполагает чтение каждой страницы книги только один раз;
- Zilele în care Andrei citește cartea sunt numerotate consecutiv, începând cu numărul 1 // Дни, в которые Андрей читает книгу, нумеруются последовательно, начиная с номера 1;
- Timpul de execuție nu va depăși 2 sec // Время выполнения программы до 2 сек;
- Programul va folosi cel mult 32 MB de memorie // Максимальный объем памяти – 32 MB.

Exemplu // Пример

INTRARE	IEȘIRE
9 7 1 3 6 8 2 4 9 5	4 2 3

Fișierul sursă va avea denumirea xxx4.PAS, xxx4.C sau xxx4.CPP, unde xxx fiind codul elevului. De exemplu, N014.PAS // Имя исходного файла будет xxx4.PAS, xxx4.C или xxx4.CPP, где xxx – регистрационный код ученика. Например, N014.PAS.

5. DRUMUL LUI FĂT-FRUMOS // ПУТЬ ГЕРАКЛА

Făt-Frumos trebuie să străbată un labirint cu capcane reprezentat de o matrice $n \times m$. Pentru fiecare celulă a labirintului, se cunoaște timpul în minute după care celula respectivă devine capcană. După ce o celulă devine capcană, Făt-Frumos moare dacă intră în acea celulă. Făt-Frumos pornește din colțul stânga-sus al labirintului și trebuie să ajungă în colțul dreapta-jos. El are nevoie de un minut ca să treacă dintr-o celulă într-una vecină și se poate deplasa în sus, în jos, spre stânga sau spre dreapta. Să se determine timpul minim în care poate Făt-Frumos să străbată labirintul și numărul de drumuri de timp minim, pe care le poate urma Făt-Frumos prin labirint de la intrare la ieșire, astfel încât Făt-Frumos să nu moară.

// Геракл должен пройти через лабиринт с ловушками, представленной матрицей размерностью $n \times m$. Для каждой клетки лабиринта известно время в минутах, после которого клетка оказывается ловушкой. Как только клетка становится ловушкой, Геракл умирает, если входит в эту клетку. Геракл начинает с верхнего левого угла лабиринта и должен попасть в правый нижний угол. От одной клетки до соседней требуется минута, и он может двигаться вверх, вниз, влево или вправо. Определить минимальное время, за которое Геракл может пройти через лабиринт, количество путей с минимальным временем по которым Геракл может пройти через лабиринт от входа до выхода и чтобы Геракл не умер.

Sarcină. Elaborați un program care determină timpul minim de parcurgere a labirintului și numărul de drumuri de timp minim.

// **Задание.** Разработайте программу, которая определяет минимальное время прохождения лабиринта и количество путей с минимальным временем.

Date de intrare. Pe prima linie se scrie numărul natural n . Pe urm.-le n linii – matricea labirintului. // **Входные данные:** В первой строке натуральное число n . В следующих n строках – матрица лабиринта.

Date de ieșire. Pe prima linie se înscriu 2 numere: timpul minim și numărul de drumuri de timp minim. Dacă nu-s drumuri se va afișa 0. // **Выходные данные:** В первой строке 2 числа: минимальное время и количество минимальных по времени путей. Если нет путей прохода - вывести 0.

Restricții // Ограничения:

- $1 \leq n \leq 20, 1 \leq m \leq 20, 0 \leq p \leq 1000, n, m \in \mathbb{N}; p \in \mathbb{R}$;
- Timpul de execuție nu va depăși 3 sec // Время выполнения программы – до 3 сек.

Exemple // Примеры

INTRARE	IEȘIRE	Explicații
6 6 3 4 5 6 7 8 3 1 1 1 1 9 5 6 7 12 11 10 1 7 1 13 1 1 1 8 1 14 15 16 1 9 10 11 12 17	11 4	Drumurile minime // Минимальные пути: 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 2 0 0 0 0 2 0 0 0 0 0 2 0 0 0 0 2 0 0 0 0 3 4 5 6 0 3 4 5 6 0 3 4 5 6 0 3 4 0 0 0 0 0 7 0 0 0 0 7 0 0 0 0 7 0 0 0 5 0 0 0 0 0 8 9 10 0 0 8 9 0 0 0 8 0 0 0 6 0 0 0 0 0 0 0 11 0 0 0 10 11 0 0 0 9 10 11 0 7 8 9 10 11
5 5 1	0	Peste 1 min toate celulele devin capcane // Через одну минуту все клетки становятся ловушками.

11111		
-------	--	--

Fișierul sursă va avea denumirea *xxx5.PAS*, *xxx5.C* sau *xxx5.CPP*, *xxx* – codul elevului. De exemplu, *N015.PAS*.

// Имя исходного файла будет *xxx5.PAS*, *xxx5.C* или *xxx5.CPP*, где *xxx* – код ученика. Например, *N015.PAS*.