

## CLASA IX – X

### 1. DRUM // ДОРОГА

Un călător își planifică o vizită a trei orașe  $A$ ,  $B$  și  $C$ , pornind din oricare din orașele  $A$  sau  $B$  și ajungând, în final, în oricare din orașele  $B$  sau  $C$  cu trecere prin cel de-al treilea oraș. Distanța în km dintre orașul  $A$  și orașul  $B$  este păstrată în variabila  $c$ , distanța în km dintre orașul  $B$  și orașul  $C$  – în variabila  $a$ , iar distanța în km dintre orașul  $C$  și orașul  $A$  – în  $b$ .

// Путешественник планирует визит трех городов  $A$ ,  $B$  и  $C$ , начиная с любого из городов  $A$  или  $B$  и достигая, в конце дороги, любой из городов  $B$  или  $C$ , пройдя через третий город. Расстояние в км между городами  $A$  и  $B$  хранится в переменной  $c$ , расстояние в км между городами  $B$  и  $C$  – в переменной  $a$ , а расстояние в км между городами  $A$  и  $C$  – в переменной  $b$ .

**Sarcină // Задание.** Alcătuiți un program, care să determine un traseu de lungime minimă, ce respectă aceste condiții. // Составить программу определения минимальной длины маршрута, соответствующем этим условиям.

**Date de intrare // Входные данные.** Fișierul text DRUM.IN conține pe prima linie trei numere întregi  $a$ ,  $b$ ,  $c$  – distanțele în km dintre orașele, respectiv,  $B$  și  $C$ ,  $A$  și  $C$ ,  $A$  și  $B$ . // Текстовый файл DRUM.IN, conține în primei строке три целых числа  $a$ ,  $b$ ,  $c$  – расстояние в км между городами, соответственно,  $B$  и  $C$ ,  $A$  и  $C$ ,  $A$  и  $B$ .

**Date de ieșire // Выходные данные.** Fișierul text DRUM.OUT va conține cele 3 litere corespunzătoare celor trei orașe, în ordinea în care sunt vizitate. // Текстовый файл DRUM.OUT будет содержать 3 буквы, соответствующие трем городам, в порядке их посещения.

#### Restricții // Ограничения:

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

Fișierul sursă va avea denumirea DRUMxx.PAS, DRUMxx.C sau DRUMxx.CPP, xx fiind codul elevului din 2 cifre. De exemplu, DRUM01.PAS. // Имя исходного файла будет DRUMxx.PAS, DRUMxx.C или DRUMxx.CPP, xx код ученика из 2 цифр. Например, DRUM01.PAS.

#### Exemplu // Пример.

DRUM.IN	DRUM.OUT
140 125 58	BAC

### 2. DEPOZIT // СКЛАД

Nicu, știind cantitățile de marfă din  $n$  camere ale unui depozit, indicate de numerele  $C_1, C_2, \dots, C_n$  a dorit să determine, dacă această marfă poate fi transportată cu exact  $n$  camioane.

// Нику, зная количество товаров по  $n$  помещениям склада, заданные числами  $C_1, C_2, \dots, C_n$  хочет определить, если товар может быть перевезен ровно  $n$  грузовиками.

**Sarcină // Задание.** Alcătuiți un program, care să determine un grup de camere cu proprietatea că suma maximă a cantităților de marfă pe care le conțin se poate împărți exact la cele  $n$  camioane care o transportă // Составить программу определения группы помещений с тем свойством, что максимальное количество товаров, которые они содержат, могут быть перевезено в точности  $n$  грузовиками.

**Date de intrare // Входные данные.** Fișierul text DEPOZIT.IN conține pe prima linie un număr natural  $n$  – numărul de camere ale depozitului, iar pe a doua linie cantitățile de marfă din  $n$  camere, separate prin spațiu. // Текстовый файл DEPOZIT.IN conține în первой строке натуральное число  $n$  – количество помещений склада, а на второй строке  $n$  натуральных чисел – количество товаров в каждом помещении склада, разделенные пробелом.

**Date de ieșire // Выходные данные.** Fișierul text DEPOZIT.OUT va conține pe o singură linie secvența de camere în ordine crescătoare, marfa cărora este transportată cu exact  $n$  camioane. // Текстовый файл DEPOZIT.OUT будет содержать в одной строке последовательность комнат по возрастанию, товар которых транспортируется ровно  $n$  грузовиками.

#### Restricții // Ограничения.

- $1 \leq n \leq 1000$ ;  $C_1, C_2, \dots, C_n, n, C_i \in \mathbb{N}$ ,  $i = 1, 2, \dots, n$ ;
- Timpul de execuție nu va depăși 1 sec // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 MO de memorie. // Максимальный объем памяти – 32 MB.

Fișierul sursă va avea denumirea DEPOZITxx.PAS, DEPOZITxx.C sau DEPOZITxx.CPP DRUMxx.PAS, xx fiind codul elevului din 2 cifre. Ex., DEPOZIT01.PAS. // Имя исходного файла будет DEPOZITxx.PAS, DEPOZITxx.C или DEPOZITxx.CPP, xx код ученика из 2 цифр. Например, DEPOZIT01.PAS.

#### Exemplu // Пример.

DEPOZIT.IN	DEPOZIT.OUT	DEPOZIT.IN	DEPOZIT.OUT
3	1 2	5	3 4
3 6 4		4 3 6 4 4	

### 3. SET DE TENIS // ТЕННИСНЫЙ СЕТ

La o competiție de tenis jucătorii  $A$  și  $B$  joacă un set. Setul este câștigat, când unul dintre jucători câștigă 6 game-uri. Jucătorii servesc pe rând. Dacă un jucător pierde serviciul, atunci se consideră, că celălalt jucător a realizat un *break*. Sandu afirmă că după numărul de brake-uri realizate de ambii jucători, se poate determina scorul setului. Dacă scorul este egal, atunci se joacă *tie-break*. // В соревновании теннисисты  $A$  и  $B$  играют сет. Сет выигран, когда один из игроков выигрывает 6 game-ов. Игроки подают поочередно. Если теннисист теряет подачу, то считается, что другой игрок сделал *break*. Sandu утверждает, что по количеству сделанных *break*-ов обоих игроков, можно определить счет сета. Если равный счет, то играется *tie-break* (тай-брейк).

**Sarcină // Задание.** Alcătuiți un program, care determină scorul și câștigătorul setului de tenis, dacă se cunoaște numărul de brake-uri realizate și totale pentru ambii jucători, de asemenea cine primul servește. // Разработать программу для определения результата теннисного сета и победителя, если известно число реализованных *break*-ов и общее число *break*-ов обоих игроков. Также известен первый подающий.

**Date de intrare // Входные данные.** Fișierul text TENNIS.IN conține:

- pe prima linie numărul de brake-uri realizate  $RA$  și totale  $TA$  ale jucătorului  $A$ ;
- pe a doua linie – numărul de brake-uri realizate  $RB$  și totale  $TB$  ale jucătorului  $B$ ;
- pe a treia linie – litera  $A$  sau  $B$  a jucătorului, care primul servește.

// Текстовый файл TENNIS.IN conține:

- в первой строке число реализованных *break*-ов  $RA$  и общее число *break*-ов  $TA$  теннисиста  $A$ ;
- во второй строке число реализованных *break*-ов  $RB$  и общее число *break*-ов  $TB$  теннисиста  $B$ ;
- в третьей строке – букву  $A$  или  $B$  первого подающего игрока.

**Date de ieșire // Выходные данные.** Fișierul text TENNIS.OUT va conține pe o linie scorul și: // Текстовый файл TENNIS.OUT conține на одной строке счет сета и:

- litera  $A$ , dacă câștigă jucătorul  $A$  // букву  $A$ , если выигрывает теннисист  $A$ ;
- litera  $B$ , dacă câștigă jucătorul  $B$  // букву  $B$ , если выигрывает теннисист  $B$ ;
- litera  $T$ , dacă se joacă tie-break букву  $T$ , если играется тай-брейк.

#### Restricții // Ограничения:

- $1 \leq RA, RB \leq 3$ ;  $1 \leq TA, TB \leq 20$ ,  $RA, RB, TA, TB \in \mathbb{N}$ ;
- Timpul de execuție nu va depăși 1 sec // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 MO de memorie. // Максимальный объем памяти – 32 MB.

Fișierul sursă va avea denumirea TENNISxx.PAS, TENNISxx.C sau TENNISxx.CPP, xx fiind codul elevului din 2 cifre. Ex. TENNIS01.PAS. // Имя исходного файла будет TENNISxx.PAS, TENNISxx.C или TENNISxx.CPP, xx код ученика из 2 цифр. Например, TENNIS01.PAS.

**Exemplu // Пример.**

TENNIS.IN	TENNIS.OUT	TENNIS.IN	TENNIS.OUT
3 10	6 1 A	2 5	6 6 T
1 2		2 3	
A		B	

**4. PERECHI // ПАРЫ**

La preselecție la un club de dansuri s-au înscris  $n$  băieți și  $m$  fete. Se pot organiza perechi, doar dacă diferența de vârstă între un băiat și o fată este de la 6 luni la 1 an, băiatul fiind mai mare. Se presupune, că un băiat (o fată) poate forma o pereche cu mai multe fete (băieți). // На этапе претестирования в танцевальном клубе записались  $n$  ребят и  $m$  девушек. Пары могут быть организованы только если разница в возрасте между парнем и девушкой от 6 месяцев до 1 года, причем парень старше. Предполагается, что парень (девушка) может образовать пару с большим количеством девушек (парней).

**Sarcină // Задание.** Alcătuiți un program, care determină numărul de perechi posibile și perechile care pot fi formate. // Составить программу определения числа возможных пар и какие пары могут быть сформированы.

**Date de intrare // Входные данные.** Fișierul text PERECHI.IN conține  $n+m+1$  linii. Pe prima linie sunt înscrise două numere naturale separate prin spațiu  $n$  și  $m$ . Pe următoarele  $n$  linii datele băieților: numele și data de naștere, formată din ziua, luna și anul nașterii separate prin spațiu. Pe următoarele  $m$  linii datele fetelor: numele și data de naștere, formată din ziua, luna și anul nașterii separate prin spațiu.

// Текстовый файл PERECHI.IN, содержит  $n+m+1$  строк. В первой строке записаны два целых числа  $n$  и  $m$ , разделенные пробелом. Следующие  $n$  строк мальчиков содержат данные для  $n$  парней: имя и дата рождения: день, месяц и год рождения, разделенные пробелом. Следующие  $m$  строк мальчиков содержат данные для  $m$  девушек: имя и дата рождения: день, месяц и год рождения, разделенные пробелом.

**Date de ieșire // Выходные данные.** Fișierul text PERECHI.OUT va conține pe prima linie un număr natural care reprezintă numărul de perechi, ce pot fi formate. Pe următoarele linii se înscriu numele băiatului și a fetei separate prin spațiu. Perechile formate se înscriu pe câte o linie separată. // Текстовый файл PERECHI.OUT в первой строке содержит натуральное число, представляющее количество пар, которые могут быть сформированы. На следующих строках выводятся имена парня и девушки каждой сформированной пары, разделенные пробелом. Сформированные пары записываются в отдельных строках.

**Restricții // Ограничения.**

- $1 \leq n, m \leq 10, n, m \in \mathbb{N}$ ;
- Timpul de execuție nu va depăși 1 sec // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 MO de memorie. // Максимальный объем памяти – 32 MB.

**Exemplu // Пример.**

PERECHI.IN	PERECHI.OUT
3 4	3
Ion 27 2 2000	Ion Maria
Alex 20 10 2000	Alex Anna
Radu 5 8 2001	Radu Alina
Maria 15 1 2001	
Elena 15 1 2002	
Anna 15 6 2001	
Alina 15 6 2002	

Fișierul sursă va avea denumirea PERECHIxx.PAS, PERECHIxx.C sau PERECHIxx.CPP, xx fiind codul elevului din 2 cifre. Ex. PERECHI01.PAS. //Имя исходного файла будет PERECHIxx.PAS, PERECHIxx.C или PERECHIxx.CPP, xx код ученика из 2 цифр. Например, PERECHI01.PAS.

**5. EXPRESIE // ВЫРАЖЕНИЕ**

Se dă un șir de  $n$  numere naturale nenule  $x_1, x_2, \dots, x_n$  și un număr natural  $m$ .

// Задана последовательность  $n$  целых чисел  $x_1, x_2, \dots, x_n$  и натуральное число  $m$ .

**Sarcină // Задание.** Alcătuiți un program, care să verifice dacă valoarea expresiei

$\sqrt[m]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$  este un număr natural, în caz afirmativ, să se afișeze acest număr, descompus în factori primi.

// Составить программу определения если значение выражения

$\sqrt[m]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$  представляет собой натуральное число, если да, то вывести это число, разложенное на простые множители.

**Date de intrare // Входные данные.** Fișierul text EXPRESIE.IN conține pe prima linie două numere naturale separate prin spațiu  $n$  și  $m$ , pe linia a doua numerele  $x_1, x_2, \dots, x_n$ , separate între ele prin câte un spațiu. // Текстовый файл EXPRESIE.IN содержит в первой строке два целых числа  $n$  и  $m$ , разделенные пробелом, а во второй строке числа  $x_1, x_2, \dots, x_n$ , разделенные пробелом.

**Date de ieșire // Выходные данные.** Fișierul text EXPRESIE.OUT va conține pe prima linie cifra 0, dacă valoarea expresiei nu este un număr natural, respectiv 1, dacă este un număr natural. Dacă valoarea expresiei este un număr natural, pe următoarele linii se vor scrie perechi de forma  $P E$  ( $P$  este factor prim care apare în descompunere la puterea  $E \geq 1$ ). Aceste perechi se vor scrie în ordine crescătoare după primul număr (adică  $P$ ). // Текстовый файл EXPRESIE.OUT будет содержать в первой строке число 0, если выражение не является натуральным числом, и 1, если выражение является натуральным числом. Если значение выражения является натуральным числом, записать в следующие строки пары вида  $P E$  ( $P$  является простым множителем, который появляется в разложении выражения в степени  $E \geq 1$ ). Эти пары будут записаны в порядке возрастания первого числа (т.е.  $P$ ).

**Restricții // Ограничения:**

- $1 \leq n \leq 5000, n \in \mathbb{N}$ ;
  - $1 \leq m \leq 10, m \in \mathbb{N}$ ;
  - $1 \leq x_i \leq 30000, x_i \in \mathbb{N}, i=1,2,\dots,n$ ;
  - Timpul de execuție nu va depăși 1 sec // Время выполнения программы – до 1 сек;
  - Programul va folosi cel mult 32 MO de memorie. // Максимальный объем памяти – 32 MB.
- Fișierul sursă va avea denumirea EXPRESIExx.PAS, EXPRESIExx.C sau EXPRESIExx.CPP, xx fiind codul elevului din 2 cifre. De exemplu, EXPRESIE01.PAS. // Имя исходного файла будет EXPRESIExx.PAS, EXPRESIExx.C или EXPRESIExx.CPP, xx код ученика из 2 цифр. Например, EXPRESIE01.PAS.

**Exemplu // Пример.**

EXPRESIE.IN	EXPRESIE.OUT	EXPRESIE.IN	EXPRESIE.OUT
2 4	0	2 4	1
32 81 100 19		32 81 100 18	2 8
			3 6
			5 2