

**Тimpul acordat: 180 min**

**Время: 180 мин**

## CLASA IX – X

### 1. CUMĂRĂTURĂ // ПОКУПКА

Un pix a costat  $K$  lei. La 1 septembrie prețul pixului a crescut cu  $P$  procente. Cumpărătorul are la dispoziție suma de  $S$  lei. // Ручка стоила  $K$  лей. Первого сентября стоимость ручки увеличилась ровно на  $P$  процентов. Известно, что у покупателя в наличии имеется сумма  $S$  лей.

**Sarcină // Задание.** Alcătuiți un program, care să determine câte pixuri pot fi procurate după scumpire, având la dispoziție  $S$  lei // Составьте программу, которая определяет сколько ручек можно купить на  $S$  лей после подорожания.

**Date de intrare // Входные данные.** Fișierul text CUMPAR.IN conține pe prima linie trei numere întregi pozitive:  $K$  – prețul pixului în lei până la scumpire,  $P$  – procentul de scumpire al pixului,  $S$  – suma de care dispune cumpărătorul // Текстовый файл CUMPARATURA.IN, conține în prima linie trei целых положительных числа:  $K$  – стоимость ручки в леях до подорожания,  $P$  – величина подорожания ручки в процентах,  $S$  – имеющаяся сумма денег.

**Date de ieșire // Выходные данные.** Fișierul text CUMPAR.OUT va conține pe prima linie numărul de pixuri ce pot fi procurate cu  $S$  lei (după scumpire) // Текстовый файл CUMPAR.OUT va conține în prima linie количество ручек, которые можно купить на  $S$  лей после подорожания.

#### Restricții // Ограничения:

- $1 \leq K, S \leq 10000, K, S \in \mathbb{N}$ ;
- $1 \leq P \leq 100, P \in \mathbb{N}$ ;
- Тimpul 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.

Fișierul sursă va avea denumirea xxx1.PAS, xxx1.C sau xxx1.CPP, unde xxx fiind codul de înregistrare a elevului format din litera N/Z și 2 cifre. De exemplu, N011.PAS. // Имя исходного файла будет xxx1.PAS, xxx1.C или xxx1.CPP, где xxx – регистрационный код ученика, сформированный из буквы N/Z и двух цифр. Например, N011.PAS.

#### Exemplu // Пример.

CUMPAR.IN	CUMPAR.OUT
33 5 100	2

### 2. PREGĂTIM VALIZA // ПАКУЕМ ЧЕМОДАНЫ

Elena pregătește valiza pentru o călătorie. În salonul avionului ea poate lua un rucsac, iar în compartimentul bagaje e posibil de depus și o valiză mare. Conform restricțiilor greutatea bagajului ce poate fi luat în salon nu poate depăși  $S$  kg, dar valiza mare poate avea orice greutate (surplusul de bagaj Elena e dispusă să-l achite). Evident, că lucrurile prețioase – laptop, aparatul de fotografiat, acte etc. Elena ar dori să le ia în salon. Ea procedează în felul următor: aranjează lucrurile personale în ordinea scăderii importanței acestora, și dacă greutatea lucrului personal nu depășește greutatea  $S$ , îl pune în rucsac, în caz contrar îl pune în valiză. Apoi ia următorul lucru și dacă greutatea lui, împreună cu cea

din rucsac nu depășește  $S$ , îl pune în rucsac, în caz contrar, în valiză, procedeul se repetă pentru toate lucrurile din șir.

// Алёна собирает вещи в отпуск. С собой в самолёт она может взять ручную кладь и багаж. Для ручной клади у Алёны есть рюкзак, а для багажа – огромный чемодан. По правилам перевозки масса ручной клади не должна превосходить  $S$  кг, а багаж может быть любой массы (за сверхнормативный багаж Алёна готова доплатить). Разумеется, наиболее ценные вещи – ноутбук, фотоаппарат, документы и т.д. – Алёна хочет положить в ручную кладь. Алёна разложила все свои вещи в порядке уменьшения их ценности и начинает складывать наиболее ценные вещи в рюкзак. Она действует следующим образом – берёт самый ценный предмет, и если его масса не превосходит  $S$ , то кладёт его в рюкзак, иначе кладёт его в чемодан. Затем она берёт следующий по ценности предмет, если его можно положить в рюкзак, то есть если его масса вместе с массой уже положенных в рюкзак вещей не превосходит  $S$ , то кладёт его в рюкзак, иначе в чемодан, и таким же образом процесс продолжается для всех предметов в порядке убывания их ценности.

**Sarcină // Задание.** Alcătuiți un program, care să determine greutatea rucsacului și a valizei, după ce Elena aranjează toate lucrurile personale // Составьте программу, которая определяет вес рюкзака и чемодана после того, как Алёна сложит все вещи.

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

- pe prima linie două numere întregi pozitive:  $S$  – greutatea admisibilă a rucsacului și  $n$  – cantitatea de lucruri personale;
- pe a doua linie – greutățile lucrurilor personale  $G_i$  în ordinea descreșterii valorii acestora.

// Текстовый файл VALIZA.IN, conține:

- в первой строке два целых положительных числа:  $S$  – максимально разрешённый вес рюкзака и  $n$  – количество предметов;
- во второй строке – массы предметов  $G_i$ , причём сами предметы перечислены в порядке убывания ценности.

**Date de ieșire // Выходные данные.** Fișierul text VALIZA.OUT va conține pe prima linie două numere – greutatea lucrurilor din rucsac și greutatea lucrurilor din valiză (greutatea rucsacului și a valizei nu se vor lua în considerație) // Текстовый файл VALIZA.OUT va conține în prima linie два числа – вес вещей из рюкзака и вес вещей из чемодана (вес пустого рюкзака и чемодана не учитывается).

#### Restricții // Ограничения:

- $1 \leq n \leq 1000, n \in \mathbb{N}$ ;
- $1 \leq S, G_i \leq 10000, S, G_i \in \mathbb{N}, i=1, 2, \dots, n$ ;
- Тimpul 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.

Fișierul sursă va avea denumirea xxx2.PAS, xxx2.C sau xxx2.CPP, unde xxx fiind codul de înregistrare a elevului format din litera N/Z și 2 cifre. De exemplu, N012.PAS. // Имя исходного файла будет xxx2.PAS, xxx2.C или xxx2.CPP, где xxx – регистрационный код ученика, сформированный из буквы N/Z и двух цифр. Например, N012.PAS.

**Exemplu // Пример.**

VALIZA.IN	VALIZA.OUT
20 5	18 8
6 10 5 2 3	

**3. LICHIDARE DE STOCURI // РАСПРОДАЖА**

Într-un magazin este inițiată lichidarea stocurilor de marfă, astfel ca prețul oricărei mărfi a fost redus cu 25%. Inițial oricare preț era divizibil cu 4, deci prețul în urma reducerii reprezintă un număr întreg. Din greșeală toate etichetele cu prețuri s-au amestecat, iar acum trebuie să se separe prețurile noi de cele vechi // В магазине проходит тотальная распродажа – цены всех товаров снижены на 25 %. Оказалось, что первоначально все цены делились на 4, поэтому после снижения цен все цены также выражаются целым числом. По ошибке, ценники смешали вместе, и теперь нужно отделить старые ценники от новых.

**Sarcină // Задание.** Alcătuiți un program, care să determine prețurile mărfurilor după reducerea prețurilor inițiale // Составьте программу, которая определяет стоимости товаров после понижения цен.

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

- pe prima linie un număr întreg pozitiv:  $n$  – numărul de etichete de preț;
- pe a doua linie – prețurile mărfurilor în ordinea crescătoare, scrise pe toate etichetele de preț.

// Текстовый файл STOC.IN, содержит:

- в первой строке целое положительное число:  $n$  – общее количество ценников;
- во второй строке – стоимости товаров в порядке их возрастания, записанные на всех ценниках (как старых, так и новых).

Este garantată existența soluției // Гарантируется, что входные данные корректны, то есть решение существует.

**Date de ieșire // Выходные данные.** Fișierul text STOC.OUT va conține pe o singură linie  $n/2$  valori întregi în ordine crescătoare – prețurile mărfurilor după reducerea prețurilor inițiale // Текстовый файл STOC.OUT будет содержать в одной строке  $n/2$  целых чисел в порядке возрастания – стоимости товаров после понижения цен.

**Restricții // Ограничения:**

- $1 \leq n \leq 1000$ ,  $n \in \mathbb{N}$ ;
- $1 \leq S_i \leq 100000$ ,  $S_i \in \mathbb{N}$ ,  $i=1, 2, \dots, n$ ;
- 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.

Fișierul sursă va avea denumirea xxx3.PAS, xxx3.C sau xxx3.CPP, unde xxx fiind codul de înregistrare a elevului format din litera N/Z și 2 cifre. De exemplu, N013.PAS. // Имя исходного файла будет xxx3.PAS, xxx3.C или xxx3.CPP, где xxx – регистрационный код ученика, сформированный из буквы N/Z и двух цифр. Например, N013.PAS.

**Exemplu // Пример.**

LICHIDARE.IN	LICHIDARE.OUT
6	30 42 45
30 40 42 45 56 60	

**4. AMPLASAMENTUL PARTICIPANȚILOR // РАССАДКА УЧАСТНИКОВ**

La o olimpiadă de informatică au sosit  $n$  participanți. Se cunosc școlile în care învață fiecare participant. În clasa de informatică sunt  $n$  calculatoare, amplasate liniar de-a lungul unui perete // На олимпиаду по информатике пришло  $n$  участников. Известно, в каких школах учатся участники олимпиады. В компьютерном классе имеется  $n$  компьютеров, стоящих в линию вдоль стены.

**Sarcină // Задание.** Alcătuiți un program, care să determine posibilitatea amplasamentului participanților olimpiadei, în așa fel ca oricare doi participanți din aceeași școală să nu stea alături // Составьте программу, которая определяет возможность расположения участников олимпиады так, чтобы любые два участника из одной школы не сидели рядом.

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

- pe prima linie un număr întreg pozitiv:  $n$  – numărul de participanți;
- pe a doua linie – numerele școlilor  $P_i$ , în care învață participanții olimpiadei.  
// Текстовый файл PART.IN, содержит:
  - в первой строке целое положительное число:  $n$  – число участников олимпиады;
  - во второй строке – номера школ  $P_i$ , в которых учатся участники олимпиады.

**Date de ieșire // Выходные данные.** Fișierul text PART.OUT va conține pe prima linie cifra 0, dacă nu există nici o soluție (este imposibilă amplasarea participanților olimpiadei conform condițiilor problemei), sau cifra 1, dacă s-a găsit o soluție oarecare a amplasamentului participanților conform condițiilor problemei .

// Текстовый файл PART.OUT будет содержать в первой строке число 0, если задача не имеет решения (нельзя рассадить участников олимпиады в компьютерном классе согласно условию задачи), или число 1, если найдено хотя бы одно решение по расположению участников олимпиады согласно условию задачи.

**Restricții // Ограничения:**

- $2 \leq n \leq 100$ ,  $n \in \mathbb{N}$ ;
- $1 \leq P_i \leq 1000$ ,  $P_i \in \mathbb{N}$ ,  $i=1, 2, \dots, n$ ;
- 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.

Fișierul sursă va avea denumirea xxx4.PAS, xxx4.C sau xxx4.CPP, unde xxx fiind codul de înregistrare a elevului format din litera N/Z și 2 cifre. De exemplu, N014.PAS. // Имя исходного файла будет xxx4.PAS, xxx4.C или xxx4.CPP, где xxx – регистрационный код ученика, сформированный из буквы N/Z и двух цифр. Например, N014.PAS.

**Exemplu // Пример.**

PART.IN	PART.OUT
12 1 2 1 1 4 1 2 2 3 3 4 4	1
12 1 1 1 1 1 1 1 1 2 2 3 3	0