

1. TUR // ЭКСКУРСИЯ

Compania „Star+” organizează excursii pe durata a n zile. Compania are și un premiu pentru beneficiari: zilele de sărbători sunt gratis și nu se contorizează. Se cunoaște data primei zi din excursie și durata în zile. De asemenea, se cunosc zilele – sărbători.

// Компания „Star+” организует экскурсии длительностью n дней. Компания также имеет приз для туристов: праздничные дни бесплатны и не учитываются при оплате. Известна дата первого дня поездки и продолжительность экскурсии в днях. Также известны даты выходных дней.

Sarcină. Elaborați un program, care ar determina data ultimei zi de excursie.

// **Задание.** Разработайте программу, которая определяет дату последнего дня экскурсии.

Date de intrare. Intrarea standard va conține pe prima linie – data primei zi de excursie: 3 numere, separate printr-un spațiu; pe a doua linie – 2 numere separate prin spațiu – n – durata excursiei și m – numărul de zile de sărbătoare; pe următoarele m linii: câte 3 numere, care semnifică zilele de sărbătoare, separate printr-un spațiu.

// **Входные данные.** Стандартный ввод содержит в первой строке – дату первого дня экскурсии: 3 числа, разделенные пробелом; во второй строке – 2 числа, разделенные пробелом – n – продолжительность поездки и m – количество праздничных дней; в следующих m строках: по 3 числа, обозначающие праздничные дни, разделенные пробелом.

Date de ieșire. Ieșirea standard va conține o linie – ultima zi de excursie: 3 numere, separate printr-un spațiu.

// **Выходные данные.** Стандартный вывод должен содержать в первой строке – дату последнего дня экскурсии: 3 числа, разделенные пробелом.

Restricții // Ограничения:

- Zerourile nesemnificative nu se afișează // незначащие нули не отображать;
- $1 \leq n \leq 100$; $0 \leq m \leq 20$;
- Se cercetează cazurile, când anul este bisect, și când perioada excursiei include trecerea dintr-un an în altul // исследовать случаи, когда год является високосным и когда период экскурсии включает переход из одного года в другой;
- Timpul de execuție nu va depăși 1 sec // время выполнения программы не будет превышать 1 сек.

Exemplu // Пример

INTRARE	IEȘIRE	Explicații // Пояснения
29 12 2019 15 3 1 1 2020 7 1 2020 14 1 2020	15 1 2020	Din luna decembrie se contorizează 3 zile: 29, 30 și 31. Din luna ianuarie, 2020 se contorizează 12 zile rămase: 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 și 15. Zilele de 1, 7 și 14 ianuarie nu se contorizează. // В декабре оплачиваются 3 дня: 29, 30 и 31. В январе, 2020 года оплачиваются только 12 дней: 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 и 15. Дни 1, 7 и 14 января не оплачиваются.

Fișierul sursă va avea denumirea $TUR_{xxx}.PAS$ sau $TUR_{xxx}.CPP$, xxx fiind codul elevului. De exemplu, $TURU01.PAS$ // Исходный файл будет иметь имя $TUR_{xxx}.PAS$ или $TUR_{xxx}.CPP$, где xxx – код ученика. Например, $TURU01.PAS$.

2. ANAGrame// АНАГРАММЫ

Este dat un cuvânt s (cu cel mult 12 caractere litere mici distincte).

// Дано слово s (длиной не более 12 символов).

Sarcină. Elaborați un program, care să determine toate anagramele cuvântului s în care consoanele sunt puncte fixe. // **Задание.** Разработайте программу, которая определяет все анаграммы слова s , в которых согласные не меняют своё положение.

Date de intrare: Intrarea standard va conține, pe prima linie cuvântul s .

// **Входные данные.** Стандартный ввод содержит в первой строке – слово s .

Date de ieșire: Ieșirea standard va conține două linii. Pe prima linie – numărul total de anagrame posibile ale cuvântului s . Pe următoarea linie – anagramele cuvântului s , separate printr-un spațiu. Anagramele se înscriu în ordine alfabetică.

// **Выходные данные.** Стандартный вывод должен содержать в первой строке – число – общее количество возможных анаграмм слова s . В следующей строке – анаграммы слова, разделенные пробелом. Анаграммы записывать в алфавитном порядке.

Restricții // Ограничения:

- Timpul de execuție nu va depăși 1 sec // Время выполнения программы не будет превышать 1 сек.

Exemplu // Пример

INTRARE	IEȘIRE	Explicații // Пояснения
alinus	6 alinus alunis ilanus ilunas ulanis ulinas	Deoarece sunt 3 vocale, în total vor fi 6 variante de permutări a acestora. // Поскольку в слове 3 гласных, то будет всего 6 вариантов их перестановок.

Fișierul sursă va avea denumirea *ANAGxxx.PAS* sau *ANAGxxx.CPP*, *xxx* fiind codul elevului. De exemplu, *ANAGU01.PAS* // Исходный файл будет иметь имя *ANAGxxx.PAS* или *ANAGxxx.CPP*, где *xxx* - код ученика. Например, *ANAGU01.PAS*.

3. ACHITAREA SERVICIILOR COMUNALE // ОПЛАТА КОММУНАЛЬНЫХ УСЛУГ

Pentru achitarea serviciilor comunale s-a format un rând din n persoane. La casierie lucrează m operatori la ghișee, numerotate de la 1 la m . Se cunosc timpurile t_i , $i = 1, 2, \dots, n$ de deservire a persoanelor de către operatori. Deservirea începe de la operatorul numărul 1, apoi operatorul numărul 2, ..., operatorul numărul m , apoi ordinea deservirii depinde dacă operatorul este liber. Se alege întotdeauna operatorul liber cu numărul mai mic.

//Для оплаты коммунальных услуг сформировалось очередь из n людей. В кассе работают m операторов за стойками, пронумерованных от 1 до m . Известно время t_i , $i = 1, 2, \dots, n$ обслуживания людей операторами. Обслуживание людей начинается с оператора под номером 1, далее обслуживает оператор под номером 2, ..., оператор под номером m , затем порядок обслуживания зависит от того, свободен ли оператор. Всегда для обслуживания выбирается свободный оператор с наименьшим номером

Sarcină. Elaborați un program, care să determine numerele operatorilor, care au deservit cele n persoane.

// **Задание.** Разработайте программу, которая определяет номера операторов, обслуживших n людей.

Date de intrare. Intrarea standard va conține două linii: pe prima linie 2 numere naturale: n (numărul de persoane) și m (numărul de operatori), separate printr-un spațiu; pe a doua linie – timpurile t_i , $i = 1, 2, \dots, n$ de deservire a persoanelor, separate printr-un spațiu.

// **Входные данные.** Стандартный ввод содержит две строки: в первой строке содержатся 2 натуральных числа: n и m , разделенных одним пробелом; во второй строке – время t_i , $i = 1, 2, \dots, n$ обслуживания людей, разделенных одним пробелом.

Date de ieșire. Ieșirea standard va conține o linie cu n numere naturale, separate printr-un spațiu: numerele operatorilor, care au deservit cele n persoane.

// **Выходные данные.** Стандартный вывод должен содержать n натуральных чисел, разделенных одним пробелом: номера операторов, которые обслужили n людей.

Restricții // Ограничения:

- $1 \leq n \leq 100$; $1 \leq m \leq 20$; $1 \leq t_i \leq 60$;
- Timpul de execuție nu va depăși 2 sec // Время выполнения программы до 2 сек.

Exemplu // Пример

INTRARE	IEȘIRE	Explicații // Пояснения
5 3 5 1 2 3 2	1 2 3 2 3	Prima persoană este deservită de operatorul 1; a doua – de operatorul 2; a treia – de operatorul 3; a patra persoană este deservită de operatorul 2; deoarece operatorul 1 încă este ocupat; a cincea – de operatorul 3, deoarece operatorii 1 și 3 încă sunt ocupați. // Первый человек будет обслужен оператором под номером 1; второй – оператором 2; третий – оператором 3; четвертый – оператором 2, так как оператор 1 пока еще занят; пятый – оператором 3, поскольку операторы 1 и 3 пока еще заняты.
5 3 5 4 5 5 5	1 2 3 2 1	Prima persoană este deservită de operatorul 1; a doua – de operatorul 2; a treia – de operatorul 3; a patra persoană este deservită de al operatorul 2; deoarece operatorul 1 încă este ocupat; a cincea – de operatorul 1, deoarece liberi sunt operatorii 1 și 3 și se alege întotdeauna operatorul liber cu numărul mai mic. // Первый человек будет обслужен оператором под номером 1; второй – оператором 2; третий – оператором 3; четвертый – оператором 2, так как оператор 1 пока еще занят; пятый – оператором 1, поскольку операторы 1 и 3 свободны и всегда выбирается свободный оператор с наименьшим номером.

Fișierul sursă va avea denumirea *ACHITxxx.PAS* sau *ACHITxxx.CPP*, *xxx* fiind codul elevului. De exemplu, *ACHITU01.PAS* // Имя исходного файла будет *ACHITxxx.PAS* или *ACHITxxx.CPP*, где *xxx* – регистрационный код ученика. Например, *ACHITU01.PAS*.

4. BILIARD // БИЛЬЯРД

O masă de biliard poate fi reprezentată printr-o matrice cu n linii și m coloane (m impar), numerotate începând cu 1. Masa de biliard are în total 6 găuri plasate în cele 4 colțuri și la mijlocul liniei 1 ($1, m/2+1$), precum și la mijlocul liniei n ($n, m/2+1$). Din poziția 1,1 se lansează o bilă la 45 de grade (spre sud-est).

Бильярдный стол может быть представлен матрицей с n строками и m столбцами (m нечетно), пронумерованных, начиная с 1. В бильярдном столе всего 6 лунок, расположенных в 4 углах и в середине строки 1 ($1, m/2+1$), а также в середине строки n ($n, m/2+1$). Начиная с позиции 1,1, один мяч запускается под углом 45 градусов (юго-восток).

Sarcină. Elaborați un program, care să determine în care dintre găuri va intra bila și după câți pași. Pașii sunt considerați numărul de poziții ale matricei prin care trece bila. În momentul în care bila atinge o margine a mesei, bila ricoșează simetric.

// **Задание.** Разработайте программу, которая определяет, в какую лунку попадет шар и за сколько шагов. Шаги - это число позиций матрицы, через которые проходит шар. Когда шар достигает края стола, он отображается симметрично.

Date de intrare. Intrarea standard va conține, pe prima linie 2 numere naturale: n și m , separate printr-un spațiu.

// **Входные данные.** Стандартный ввод содержит в первой строке 2 натуральных числа n и m , разделенные пробелом.

Date de ieșire. Ieșirea standard va conține pe prima linie 3 numere naturale: poziția găurii, în care va intra bila (linie și coloană) și după câți pași. În următoarele n linii – traseul bilei pe pași.

// **Выходные данные.** Стандартный вывод должен содержать в первой строке 3 натуральных числа: положение лунки, куда попадет шар (линия и столбец) и за сколько шагов. В следующих n строках – пошаговый путь шара.

Restricții // Ограничения:

- $2 \leq n \leq 20$; $3 \leq m \leq 19$;
- Timpul de execuție nu va depăși 1 sec. // Время выполнения программы – до 1 сек.

Exemple // Примеры

INTRARE	IEȘIRE	INTRARE	IEȘIRE
4 5	1 3 7	6 15	6 8 36
	1 0 7 0 0		1 0 31 0 0 0 0 0 21 0 11 0 0 0 0
	0 2 0 6 0		0 30 0 32 0 0 0 22 0 20 0 12 0 0 0
	0 0 3 0 5		29 0 3 0 33 0 23 0 9 0 19 0 13 0 0
	0 0 0 4 0		0 28 0 4 0 34 0 8 0 0 0 18 0 14 0
	0 0 0 4 0		0 0 27 0 25 0 35 0 0 0 0 0 17 0 15
			0 0 0 26 0 6 0 36 0 0 0 0 0 16 0

Fișierul sursă va avea denumirea *BILlxxx.PAS* sau *BILlxxx.CPP*, unde *xxx* fiind codul de înregistrare a elevului. De exemplu, *BILIU01.PAS* // Имя исходного файла будет *BILlxxx.PAS* или *BILlxxx.CPP*, где *xxx* – код ученика. Например, *BILIU01.PAS*.

5. COMPANIE // КОМПАНИЯ

Într-o companie sunt n angajați, numerotați de la 1 la n , fiecare angajat având un singur șef direct, cu excepția directorului, care nu are șef. Ierarhia firmei este dată printr-un vector de tip părinte. Fiecare angajat al firmei are un salariu dat printr-un număr natural. Angajații și șeful sunt recompensați astfel: câștigul fiecărui salariat este egal cu salariul său la care se adaugă media aritmetică a câștigurilor subordonaților săi direcți. Media aritmetică se rotunjește prin adaos la un număr întreg (de exemplu 5.33 se rotunjește la 6). Angajații care nu au subordonați direcți câștigă doar salariul.

// В некоторой компании работают n сотрудников, пронумерованные от 1 до n , причем каждый сотрудник имеет только одного непосредственного руководителя, за исключением директора, у которого нет начальника. Структура (иерархия) компании задается при помощи вектора родительского типа. Каждому сотруднику компании выдается зарплата, заданная натуральным числом. Сотрудникам и руководителю выплачивается денежное вознаграждение следующим образом: доход каждого работника равен его заработной плате, к которой прибавляется среднее арифметическое доходов его прямых подчиненных. Среднее арифметическое округляется путем добавления к целому числу (например, 5.33

округляется до 6). Сотрудники, у которых нет прямых подчиненных, получают только заработную плату.

Sarcină. Elaborați un program, care să determine care este câștigul angajaților companiei.

// **Задание.** Разработайте программу, которая вычисляет доход сотрудников компании.

Date de intrare. Intrarea standard va conține, pe prima linie un număr natural: numărul de angajați n . Pe a doua linie – n numere – vectorul părinte. Pe a treia linie – n numere – salariile inițiale respective, separate printr-un spațiu.

// **Входные данные.** Стандартный ввод содержит в первой строке одно натуральное число n : количество сотрудников компании. Во второй строке – n чисел – вектор родительского типа в виде отношений подчиненности. В третьей строке – n чисел – начальные заработные платы сотрудников.

Date de ieșire. Ieșirea standard va conține o linie cu n numere – salariile fiecărui angajat, separate printr-un spațiu. // **Выходные данные.** Стандартный вывод должен содержать одну строку с n числами – доход каждого сотрудника.

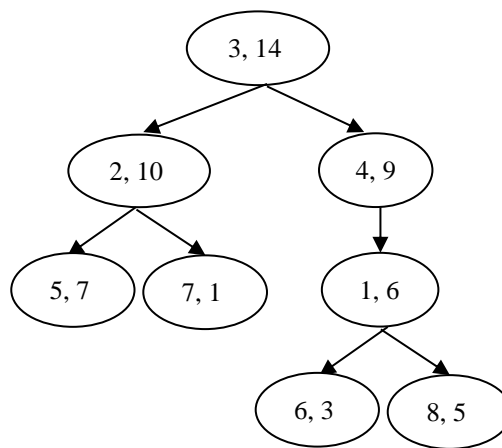
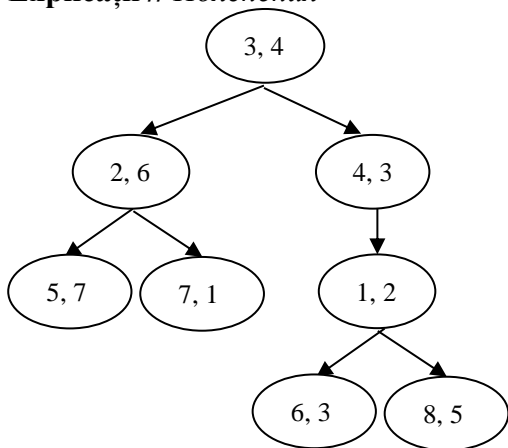
Restricții // Ограничения:

- $1 \leq n, m \leq 100$;
- Timpul de execuție nu va depăși 2 sec // Время выполнения программы до 2 сек.

Exemplu // Пример

INTRARE	IEȘIRE
8	6 10 14 9 7 3 1 5
4 3 0 3 2 1 2 1	
2 6 4 3 7 3 1 5	

Explicații // Пояснения



Fișierul sursă va avea denumirea $COMP_{xxx}.PAS$ sau $COMP_{xxx}.CPP$, xxx fiind codul elevului. De exemplu, $COMPU01.PAS$ // Имя исходного файла будет $COMP_{xxx}.PAS$ или $COMP_{xxx}.CPP$, где xxx – регистрационный код ученика. Например, $COMPU01.PAS$.