CLASA XI – XII

1. ALBUM // АЛЬБОМ

Ionel este un fan al muzicii şi vrea să descarce un album al interpretului preferat. O piesă poate fi descărcată în A secunde (timpul de descărcare depinde de volumul fişierului). Piesele (fişierele respective) pot fi descărcate consecutiv în seturi – câte una sau, în paralel, câte două sau, cel mult, câte trei. Dacă Ionel descarcă câteva fişiere în paralel, timpul de descărcare creşte: pentru două fişiere descărcate în paralel – cu 40%, iar pentru trei fişiere – cu 60%. Dacă se dă la descărcare un set de fişiere, următorul set de fişiere poate fi descărcat doar după descărcarea completă a setului precedent. Piesele sunt descărcate în ordinea lor din album. // Ваня очень любит музыку и хочет скачать альбом любимого исполнителя. Одно произведение может быть скачено за A секунд (время зависит от объема файла). Произведения (соответственно файлы) могут быть скачены последовательно по одному или параллельно по два или три файла. Если Ваня скачивает несколько файлов параллельно, то тогда время скачивания одного файла увеличивается на 40%, если скачиваются по два файла параллельно, и на 60%, если по три. Если скачивается набор файлов, то следующий набор файлов может быть скачен только после завершения скачивания всех файлов предыдущего набора. Произведения из альбома скачиваются в том же порядке, в каком они встречаются в альбоме.

Sarcină // Задание. Alcătuiţi un program, care ar determina timpul minim (în secunde), necesar pentru descărcarea albumului. // Напишите программу которая определит минимальное время (в секундах) необходимое для скачивания всего альбома.

Date de intrare // Входные данные. Fişierul text ALBUM.IN conţine pe prima linie un număr întreg N – numărul de piese, iar pe fiecare din următoarele N linii timpul de descărcare a piesei respective. // Файл ALBUM.IN содержит в первой строке целое число N – количество произведений в альбоме, а в следующих N строках время скачивания для соответствующего произведения.

Date de ieşire // Выходные данные. Fişierul text ALBUM.OUT va conţine un număr întreg, obţinut prin rotunjirea până la întregi prin adaos a valorii timpului minim (în secunde), necesar pentru descărcarea albumului. // Файл ALBUM.OUT будет содержать целое число, полученное при округлении до целого с добавлением, соответствующие минимальному времени (в секундах) необходимого для скачивания альбома.

Restricţii // Ограничения:

- $1 \leq N \leq 100$;
- Timpul de execuţie nu va depăşi 1 secundă // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 MO de memorie operativă. // Максимальный объем памяти – 32 MB.

Exemplu de date de intrare/ieşire// Пример данных ввода/вывода.

ALBUM.IN	ALBUM.OUT
4 6 8 3 7	19

Fişierul sursă va avea denumirea ALBUMxxx.PAS, ALBUMxxx.C sau ALBUMxxx.CPP, xxx fiind codul elevului din 3 cifre. Ex. ALBUM001.PAS. //Имя исходного файла будет ALBUMxxx.PAS, ALBUMxxx.C или ALBUMxxx.CPP, xxx код ученика из 3 цифр. Например, ALBUM001.PAS.

2. CIURUL LUI ERATOSTENE // РЕШЕТО ЭРАТОСФЕНА

Matematicianul grec antic Eratostene (275 - 194 î.Hr.) a aplicat o metodă inedită pentru aflarea numerelor prime. De exemplu, se scrie şirul numerelor naturale de la 2 până la 100. Luând primul număr se exclud din şir toţi multiplii acestui număr. Apoi, se alege următorul număr ne exclus şi se exclud toţi multiplii lui. Procesul se repetă până la sfârşitul şirului. // Древнегреческий математик Эратосфен (275 – 194 до. Р.Х.) использовал необычный метод для определения простых чисел. Например, запишем все числа от 2 до 100. Взяв первое число, вычеркиваем из последовательности все числа кратные выбранному. Далее выбираем следующее не зачеркнутое число и вычеркиваем из последовательности все числа кратные выбранному. Процесс повторяем до конца последовательности.

Sarcină // Задание. Alcătuiţi un program, care afişează numerele excluse prin metoda Ciurului lui Eratostene la determinarea numerelor prime mai mici ca N. // Напишите программу которая выводит все зачеркнутые числа при приминении метода Решито Эрастофена для определения простых чисел меньше заданного числа N.

Date de intrare // Входные данные. Fişierul text CIUR.IN conţine pe o singură linie cu un număr natural N. // Файл CIUR.IN содержит целое число N.

Date de ieşire // Выходные данные. Fişierul text CIUR.OUT va conţine pe prima linie multiplii primului număr prim, pe a doua linie multiplii celui de-al doilea număr prim ş.a.m.d. separate prin spaţiu şi ordonate crescător. // Файл CIUR.OUT будет содержать в первой строке все числа кратные первому простому числу, во второй строке - все числа кратные второму простому числу и т.д., разделенные пробелами и упорядоченные по возрастанию.

Restricţii // Ограничения:

- $1 \leq N \leq 1000$;
- Timpul de execuţie nu va depăşi o secundă // Время выполнения программы – до 1 сек;

Programul va folosi cel mult 32 MO de memorie operativă. // Максимальный объем памяти – 32 MB.
Fişierul sursă va avea denumirea CIUR.PAS, CIUR.C sau CIUR.CPP.

Exemplu de date de intrare/ieşire// Пример данных ввода/вывода.

CIUR.IN	CIUR.OUT
30	4 6 8 10 12 14 16 18 20 22 24 26 28 30 9 15 21 27 25

Fişierul sursă va avea denumirea CIURxxx.PAS, CIURxxx.C sau CIURxxx.CPP, xxx fiind codul elevului din 3 cifre. Ex. CIUR001.PAS. //Имя исходного файла будет CIURxxx.PAS, CIURxxx.C или CIURxxx.CPP, xxx код ученика из 3 цифр. Например, CIUR001.PAS.

3. SUMA CIFRELOR // СУММА ЦИФР

Sarcină // Задание. Alcătuiţi un program, care afişează câte numere de 9 cifre au suma cifrelor egală cu N. // Напишите программу которая определяет у скольких 9-тизначных чисел сумма цифр равна N.

Date de intrare // Входные данные. Fişierul text NUMERE.IN conţine o singură linie cu un număr natural N. // Файл NUMERE.IN содержит целое число N.

Date de ieşire // Выходные данные. Fişierul text NUMERE.OUT va conţine o linie, pe care va fi afişat numărul de numere cu suma cifrelor egală cu N. // Файл NUMERE.OUT будет содержать одну строку в которой указанно количество чисел.

Restricţii // Ограничения:

- $1 \leq N \leq 50$;
- Timpul de execuţie nu va depăşi o secundă // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 Mo de memorie operativă//Максимальный объем памяти – 32 MB.

Exemplu de date de intrare/ieşire:

NUMERE.IN	NUMERE.OUT
3	45

Fişierul sursă va avea denumirea NUMxxx.PAS, NUMxxx.C sau NUMxxx.CPP, xxx fiind codul elevului din 3 cifre. Ex. NUM001.PAS. //Имя исходного файла будет NUMxxx.PAS, NUMxxx.C или NUMxxx.CPP, xxx код ученика из 3 цифр. Например, NUM001.PAS.

4. CAMION // ГРУЗОВИК

Un camion poate transporta T tone de mobilă. Există N piese unice de mobilă (scaun, masă, dulap etc.), fiecare piesă este caracterizată de greutatea G_i şi de valoarea de câştig C_i adusă la transportarea ei. Să se decidă ce piese de mobilă vor fi transportate pentru a aduce un câştig maxim. // Один грузовик может перевозить T тонн мебели. Даны N уникальных предметов мебели (стул, стол, шкаф и т.д.), каждый предмет характеризуется величиной веса Gi и коэффициентом полезности Ci получаемый от его перевозки. Решите, какие предметы мебели нужно транспортировать, чтобы получить максимальную суммарный коэффициент полезности.

Sarcină // Задание. Alcătuiţi un program, care determină câştigul maxim posibil şi piesele care trebuie transportate. // Напишите программу которая определяет возможный максимальный суммарный коэффициент полезности и предметы мебели которые надо перевезти.

Date de intrare // Входные данные. Fişierul text CAMION.IN conţine 3 linii. Pe prima linie sunt două numere separate prin spaţiu N şi T. Pe cea de a doua linie se află numerele naturale separate prin spaţiu, care reprezintă greutăţile G_i celor N obiecte. Pe cea de a treia linie se află numerele naturale separate prin spaţiu, care reprezintă valoarea de câştig C_i corespunzătoare celor N obiecte. // // Файл CAMION.IN содержит 3 строки. В первой строке даны два целых чисел N и T, разделенные пробелом. Во второй строке заданы целые числа разделенные пробелом, которые соответствуют весу G_i предметам мебели. В третьей строке заданы целые числа разделенные пробелом, которые соответствуют коэффициенту полезности Ci предметов мебели.

Date de ieşire // Выходные данные. Fişierul text CAMION.OUT va conţine pe prima linie un număr natural care reprezintă valoarea maximă de câştig obţinută. Pe cea de a doua linie vor fi scrise indicele obiectelor selectate pentru a obţine acest câştig maxim separate prin spaţiu, ordonate crescător după indice. // Файл CAMION.OUT будет содержать в первой строке целое - максимальный суммарный коэффициент полезности. Во второй строке будут записаны индексы предметов мебели которые надо перевезти для получения этого максимального суммарного коэффициента полезности, разделенные пробелом и упорядоченные по возрастанию.

Restricţii // Ограничения:

- $1 \leq N \leq 100$;
- $1 \leq T \leq 300$;
- $G_i, C_i \leq 100$;
- Timpul de execuţie nu va depăşi o secundă // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 Mo de memorie operativă // Максимальный объем памяти – 32 MB.

Exemplu de date de intrare/ieşire:

CAMION.IN	CAMION.OUT
7 100 80 10 20 10 100 30 55 8 20 5 1 10 30 20	70 2 6 7

Fişierul sursă va avea denumirea CAMxxx.PAS, CAMxxx.C sau CAMxxx.CPP, xxx fiind codul elevului din 3 cifre. Ex. CAM001.PAS. //Имя исходного файла будет CAMxxx.PAS, CAMxxx.C или CAMxxx.CPP, xxx код ученика из 3 цифр. Например, CAM001.PAS.

5. POLIGON CONVEX // ВЫПУКЛЫЙ МНОГОУГОЛЬНИК

Fie este dată o mulţime de N puncte în plan cu coordonatele (x_i, y_i). Să se determine dacă punctele mulţimii date pot fi vârfurile unui poligon convex. Poligonul este convex dacă are proprietatea de a fi în întregime situat de aceeaşi parte a dreptei determinate de oricare din laturile lui.

// Дано множество точек N, заданных при помощи координат (x_i, y_i). Необходимо определить если эти точки могут быть вершинами выпуклого многоугольника. Выпуклым многоугольником называется многоугольник, все точки которого лежат по одну сторону от любой прямой, проходящей через две его соседние вершины.

Sarcină // Задание. Alcătuiţi un program, care determină dacă punctele mulţimii pot fi vîrfurile unui poligon convex. // Напишите программу которая определяет если точки множества N могут быть вершинами выпуклого многоугольника.

Date de intrare // Входные данные. Fişierul text POLIVEX.IN conţine pe prima linie un număr natural N care reprezintă numărul de puncte în plan. Pe următoarele N linii se află coordonatele x_i, y_i a punctelor, separate prin spaţiu. // Файл POLIVEX.IN содержит в первой строке целое число N - количество точек. В последующих N строчках содержатся координаты x и y соответствующей точки разделенные пробелом.

Date de ieşire // Выходные данные. Fişierul text POLIVEX.OUT va conţine pe prima linie 1, dacă punctele date pot fi vârfurile măcar ale unui poligon convex, 0 – în caz contrar. // POLIVEX.OUT будет содержать 1 если точки могут быть вершинами выпуклого многоугольника и 0 – если нет.

Restricţii // Ограничения:

- $1 \leq N \leq 100$;
- $-1000 \leq x, y \leq 1000$;
- Timpul de execuţie nu va depăşi o secundă // Время выполнения программы – до 1 сек;
- Programul va folosi cel mult 32 Mo de memorie operativă // Максимальный объем памяти – 32 MB.

Exemplu de date de intrare/ieşire:

POLIVEX.IN	POLIVEX.OUT
4 0 0 0 2 2 0 3 3	1

Fişierul sursă va avea denumirea POLxxx.PAS, POLxxx.C sau POLxxx.CPP, xxx fiind codul elevului din 3 cifre. Ex. POL001.PAS. //Имя исходного файла будет POLxxx.PAS, POLxxx.C или POLxxx.CPP, xxx код ученика из 3 цифр. Например, POL001.PAS.