

Задачи заочной олимпиады Программирование

Задача 1

Число m

Входной файл: input.txt

Выходной файл: output.txt

Для данного числа m найти наименьшее положительное целое число с суммой цифр m , которое делится на m .

Ввод

Во входном файле записано число m ($1 \leq m \leq 1000$).

Вывод

Выходной файл содержит искомое число. Ведущие нули выводить не разрешается.

Пример:

Пример ввода	Пример вывода
1	1
10	190

Задача 2

Факториал

Входной файл: input.txt

Выходной файл: output.txt

Выражение $N!$ (читается как “ N факториал”) означает произведение всех целых чисел от 1 до N включительно, где N – положительное число. (По соглашению $0!=1$).

Так например,

N	N!
0	1
1	1
2	2
3	6
4	24
5	120
10	3628800

Требуется написать программу, которая может вычислить последнюю, не равную нулю цифру для любого факториала ($0 \leq N \leq 10000$). Например, для числа 5 программа должна выдать 2, поскольку $5!=120$, и 2 – последняя ненулевая цифра в числе 120.

Ввод

Во входном файле input.txt записана серия неотрицательных целых чисел, не превышающих 10000, каждое число в своей собственной строке.

Вывод

В выходной файл output.txt для каждого целого должна содержаться точно одна строка. Каждая строка должна содержать величину N , в позициях с 1 по 5 с ведущими пробелами и не ведущими нулями. Позиции с 6 по 9 должны содержать “->”. Позиция 10 должна содержать единственную последнюю не равную нулю цифру $N!$.

Пример ввода	Пример вывода
1	1 -> 1
2	2 -> 2
26	26 -> 4
125	125 -> 8
3125	3125 -> 2
9999	9999 -> 8

Задача 3 Про треугольники

Входной файл input.txt

Выходной файл output.txt

Найти количество различных невырожденных треугольников таких, что их периметр равен N.

Ввод

В первой строке целое число N ($1 \leq N \leq 320000$).

Вывод

Количество треугольников.

Пояснения к примеру: существует три невырожденных треугольника с периметром N (4,3,2), (4,1,4) и (3,3,3).

Пример ввода	Пример вывода
9	3

Задача 4 Финансы

Входной файл: input.txt

Выходной файл: output.txt

Некто окончил университет, нашел работу и решил, что ему нужно вести учет его накоплений, чтобы решать финансирование своих проблем. У него имеется счет в банке и он может видеть сколько денег у него осталось за каждый месяц. Помогите ему, написать программу, которая подводит закрытие баланса по каждому из прошлых двенадцати месяцев и вычисляет его средний баланс за учетный период.

Ввод

Вклад будет состоять из двенадцати строк. Каждая строка будет содержать заключительный баланс его счета в банке в течение конкретного месяца. Каждое число будет положительным с точностью до пенни. Никакой знак обозначения доллара не будет включен.

Вывод

Выходом будет единственное число, среднее число (средство) закрытия баланса в течение двенадцати месяцев. Оно будет округлено до ближайшего числа с точностью до пенни, ему предшествует знак доллара.

Пример ввода	Пример вывода
100.00 489.12 12454.12 1234.10 823.05	\$1581.42

109.20 5.27 1542.25 839.18 83.99 1295.01 1.75	
---	--

Задача 5

Сумма

Входной файл: input.txt

Выходной файл: output.txt

Заданы два целых числа A и B , написать программу, которая вычисляет сумму целых чисел между A и B .

Ввод

Первая строка содержит целое T , которое представляет количество тестов. В следующих T строках - одна строка на тест, содержится два целых A и B .

Вывод

Ваша программа должна печатать t целых, по одному в строке. i -целое есть ответ, который Ваша программа вычисляет для случая i -го теста.

Пример ввода	Пример вывода
2	9
4 5	14
5 2	

Задача 6

Сообщающиеся сосуды

Входной файл input.txt

Выходной файл output.txt

Имеется система резервуаров в форме параллелепипедов, сообщающихся тонкими трубочками (их собственный объём пренебрежимо мал по сравнению с объёмами резервуаров). В систему залита вода. Определить установившийся уровень воды.

Ввод

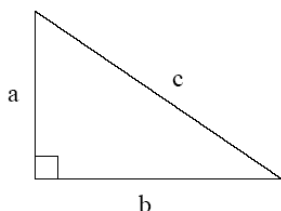
Первая строка содержит натуральное число N - количество резервуаров.

Вторая строка содержит объём воды в кубометрах.

В каждой из последующих N строк идут четвёрки натуральных чисел, разделённых пробелами. Они дают информацию о каждом резервуаре: "уровень дна, высота, длина, ширина" - все данные в метрах.

Вывод

Вещественное число - высота уровня воды (два знака после запятой) или OVERFLOW - если объём превосходит вместимость системы.



Ввод:

Входной файл содержит описания нескольких треугольников. Каждое описание состоит из строки, содержащей три целых a , b и c , задающие длины соответствующих сторон прямоугольного треугольника. Одно из трех чисел равняется точно -1 (неизвестная сторона), другие - положительное числа (длины заданных сторон).

Описание, имеющее $a=b=c=0$ означает завершение ввода.

Вывод:

Для каждого описания треугольника во входном файле, в первой строке печатается номер треугольника, как показано в примере. Затем печатается "Impossible.", если прямоугольного треугольника не существует. В противном случае печатается длина неизвестной стороны в формате $s = l$, где s - имя неизвестной стороны (a , b или c), и l - длина. l должно быть напечатано с тремя цифрами справа от десятичной точки.

После каждого варианта теста печатается пустая строка.

Пример ввода	Пример вывода
<pre>3 4 -1 -1 2 7 5 -1 3 0 0 0</pre>	<pre>Triangle #1 c = 5.000 Triangle #2 a = 6.708 Triangle #3 Impossible.</pre>

Задача 8

Очертание

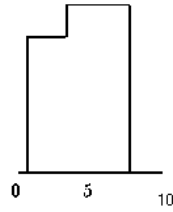
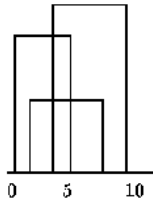
Входной файл: `input.txt`

Выходной файл: `output.txt`

Вы должны составить программу, которая помогает архитектору начертить очертание города, заданного положением зданий в городе. Для простоты, все здания прямоугольной в формы и они имеют общий низ.

Город рассматривается как двумерный. Здание определено упорядоченными тройками (L_i, H_i, R_i) , где L_i и R_i - левая и правая координаты соответственно i здания H_i - высота здания. На рисунке ниже слева показаны здания, заданные тройками $(1,11,5)$, $(2,6,7)$, $(3,13,9)$, очертание, показанное справа, представлено последовательностью:

$(1, 11, 3, 13, 9, 0)$



Ввод

Ввод является последовательностью троек здания. Все координаты зданий - положительные целые числа меньше чем 10,000 и найдется по крайней мере одно здание, а самое большее 5,000 зданий во входном файле. Каждое тройка чисел находится в своей строке во входном файле. Все числа в тройке разделены одним или более пробелами. Тройки отсортированы по L_i , левой x -координате здания, так что здания с наименьшей левой x -координатой идут первыми во входном файле.

Вывод

Выход должен состоять из вектора, который описывает очертание как показано в примере выше. В векторе очертания $(\vartheta_1, \vartheta_2, \vartheta_3, \dots, \vartheta_{n-2}, \vartheta_{n-1}, \vartheta_n)$, ϑ_i , такое, что i четное число - представляет горизонтальную линию (высоту). ϑ_i такое, что i нечетное число - представляет вертикальную линию (x -координата). Вектор очертания должен представить путь" проходящий, например, жуком, начинающимся в минимальной x -координате и путешествуя горизонтально и вертикально по всем линиями, которые определяют очертание. Таким образом, последний вход на векторе очертания будет 0. Координаты должны быть разделены пробелом.

Пример ввода	Пример вывода
<pre>1 11 5 2 6 7 3 13 9</pre>	<pre>1 11 3 13 9 0</pre>

Задача 9

Сумма двух чисел

Входной файл `input.txt`

Выходной файл `output.txt`

Заданы три числа: a , b , c . Необходимо выяснить, можно ли так переставить цифры в числах a и b , чтобы в сумме получилось c .

Ввод

Входной файл INPUT.TXT содержит три целых числа: a , b , c ($0 < a, b, c < 10^9$). Числа разделены пробелом.

Вывод

В выходной файл OUTPUT.TXT следует вывести YES, если искомая перестановка цифр возможна, в противном случае необходимо вывести NO. При положительном ответе во второй строке следует вывести число x , получаемое перестановкой цифр числа a , и число y , получаемое перестановкой цифр числа b , сумма которых равна c . Числа x и y при выводе не должны содержать ведущих нулей. Числа в строке разделены пробелом. Если решений несколько, то следует вывести ту пару, в которой число x минимально.

Пример ввода	Пример вывода
12 31 25	YES 12 13
12 31 26	NO
101 2 13	YES 11 2

Робототехника

Задача 1

Балансирование

К строго вертикальной стене прикреплен брусок длиной 200мм (см. рис.), с грузом массой 100г на конце. К стене брусок крепится через двигатель с крутящим моментом 10кг/см, который может наклонять брусок в вертикальной плоскости x,z .

Двигатель управляется с помощью программы, в которой по-умолчанию опеределены две служебные функции:

setPower(int v); - функция, которая задает мощность двигателя в процентом отношении от -100% до +100%; +100% означает, что двигатель наклоняет груз вверх с максимальным крутящим моментом.

getAngle(); - функция, которая возвращает угол наклона груза в диапазоне от -90 до +90 градусов. Функция покажет 0 градусов, в случае когда груз и брусок строго перпендикулярны стене, и -90 градусов, когда груз наклонен вниз и горизонтален стене. Программа должна удерживать груз и брусок строго в горизонтальном положении.

|
|M-----o
|

Задача 2

Прокладка маршрута

Поле размером 160м x 80м разделено на клетки размером 10м x 10м. В центре поля находится здание с дверным проемом. В разных местах поля раскиданы непроходимые камни, занимающие ровно одну клетку. На краю поля находится одинокий робот, задача которого заехать в здание, и встать в нижний левый угол. У робота нет никаких датчиков, способных определить препятствие. В программе робота определена служебная функция moveToXY(int x, int y); - функция, которая перемещает робота из текущей клетки, в клетку с координатами (x, y). Робот может перемещаться только в вертикальном и горизонтальном направлении.

На вход программы передается карта местности, в виде матрицы 16x8 клеток, в которой единицами обозначены стены, а нулями - свободное пространство. R - начальное положение робота.

Результатом работы программы является последовательность клеток, через которые должен пройти робот, не задев здания и препятствия.

```
R0000000010000000
0000000000000000
0010011111001000
0000010001000000
0000010000000000
0000011111000100
0100000000000000
0000000100000000
```

Задача 3 **Картограф**

Робот картограф, попав в неизвестное помещение, должен построить карту стен и препятствий в этом помещении. Помещение имеет размер 500м x 500м, и разбито на клетки размером 10м x 10м.

Программа робота имеет две встроенные функции:

getObstacle(); - функция, которая возвращает истину, если следующая перед роботом клетка занята, и ложь - в противном случае.

moveToXY(int x, int y) - функция, которая перемещает робота из текущей клетки, в клетку с координатами (x, y).

На вход программы передается карта местности, в виде матрицы 50x50 клеток, в которой единицами обозначены стены, а нулями - свободное пространство.

Результатом работы программы является матрица 50x50 клеток, синтезированная программой робота. Как и во входной матрице, единицами обозначены стены, а нулями - свободное

Справочный материал

Поскольку часто именно файловый ввод-вывод является основной проблемой начинающих участников олимпиад, мы подготовили небольшую памятку, демонстрирующую реализацию ввода-вывода на различных языках программирования.

Программа на любом языке программирования должна завершаться с кодом возврата 0. В частности, программа на C всегда должна заканчиваться оператором return 0; Имена входных и выходных файлов должны быть написаны маленькими латинскими буквами. Не забывайте закрывать выходной файл (иначе во многих языках он остается пустым).

Ввод-вывод будем демонстрировать на примере двух задач. Первая задача - в файле sum.in записаны два целых числа по модулю не превышающих 32000, в файл sum.out нужно вывести их сумму. Вторая задача - такая же, только числа не превышают 10^{10} (и не помещаются в 32-битный тип данных).

Pascal, Delphi

Примечание для программирующих на Delphi: Имена стандартных модулей пишутся точно так же, как в документации (имеется в виду сочетание больших и маленьких букв) и только так. Например: Math, SysUtils.

Первая задача	Вторая задача	Примечания
<pre>var a,b:longint; begin assign(input,'sum.in'); reset(input); assign(output,'sum.out'); rewrite(output); read(a,b); writeln(a+b); close(input); close(output); end.</pre>	<pre>var a,b:int64; begin assign(input,'sum.in'); reset(input); assign(output,'sum.out'); rewrite(output); read(a,b); writeln(a+b); close(input); close(output); end.</pre>	<p>Тип int64 существует в Delphi и Free pascal и отсутствует в Borland pascal.</p>

GNU C

Первая задача	Вторая задача	Примечания
<pre>#include <stdio.h> int main(void) { long a, b; FILE *fin, *fout; fin = fopen("sum.in", "r"); fout = fopen("sum.out", "w"); fscanf(fin, "%ld%ld", &a, &b); fprintf(fout, "%ld", a + b); fclose(fin); fclose(fout); return 0; }</pre>	<pre>#include <stdio.h> int main(void) { long long a, b; FILE *fin, *fout; fin = fopen("sum.in", "r"); fout = fopen("sum.out", "w"); #ifdef WIN32 fscanf(fin, "%I64d%I64d", &a, &b); #else fscanf(fin, "%lld%lld", &a, &b); #endif #ifdef WIN32 fprintf(fout, "%I64d\n",a+b); #else fprintf(fout, "%lld\n",a+b); #endif fclose(fin); fclose(fout); return 0; }</pre>	<p>Примечание для программирующих на GNU C/C++: При выводе значений переменных типа long long под ОС Windows нужно использовать спецификатор %I64d, а при сдаче решений на проверку (для ОС Linux) - спецификатор %lld. "Универсальный" код, который работает правильно под обеими системами, может выглядеть так: <pre>#ifdef WIN32 printf("%I64d\n", ans); #else printf("%lld\n", ans); #endif</pre></p>

GNU C++

Первая задача	Вторая задача
<pre>#include <fstream> using namespace std; int main () { long a, b; ifstream fin ("sum.in"); ofstream fout ("sum.out"); fin >> a >> b; fout << a + b << endl; fin.close (); fout.close (); return 0; }</pre>	<pre>#include <fstream> using namespace std; int main () { long long a, b; ifstream fin ("sum.in"); ofstream fout ("sum.out"); fin >> a >> b; fout << a + b << endl; fin.close (); fout.close (); return 0; }</pre>

Желаем успеха!