



XIV открытая региональная межпредметная олимпиада «Золотая середина» с участием стран СНГ

Разбор заданий

Информатика

Программирование на Pascal/Python

Максимальное количество баллов: 600

Можно использовать стандартные потоки ввода/вывода (с клавиатуры), либо файлы *input.txt* и *output.txt*

Ограничение по времени: 2 секунды на тест

Ограничение по памяти: 512 МБ

Авторские решения в данном документе приведены на языках PascalABC.NET и Python 3

...и однажды, после долгих испытаний, вы обнаружите, что это не так ужасно, как выглядит на самом деле.

Ричард Фейнман

1. Невероятно сложная задача

Условие задачи

Байтляндия — мирное государство, уважающее науку, именно поэтому количество научно-исследовательских институтов растёт с каждым годом. Однако научный персонал не успевает за прогрессом, следовательно, им нужна ваша помощь в автоматизации расчётов. Они вывели невероятно сложную формулу, в которой был скрыт смысл жизни и всего остального. Ваша задача — написать программу, которая посчитает значение функции $f(x)$ при заданном x .

$$f(x) = p\left((1985 - 1984) + \left(5 - \frac{25}{5}\right) \cdot \arctg(\sqrt{14 \cdot e^{\sqrt{\pi x}} + \sin(2^{e \cdot \sqrt{\frac{\pi}{2}} \cdot x e^x})})\right)$$
$$p(x) = 38 \cdot x + 4$$

Формат входных данных

Единственное натуральное число x ($1 \leq x \leq 10^{18}$).

Формат выходных данных

Единственное целое число – результат расчётов.

Категоризация

Умение читать формальное описание задачи и замечать детали, знание понятия «функция» и стандартных форм её записи.

Задание считается лёгким, но ученикам 6 класса и младше пригодились бы базовые знания математики 7-8 классов о функциях, которые можно получить в учреждениях дополнительного образования или самостоятельно.

Алгоритм решения

Из названия задачи можно предположить, что на самом деле ничего страшного в ней нет, и надо искать особенности.

Преобразуем функцию $f(x)$ в два простых шага.

$$f(x) = p\left((1985 - 1984) + (5 - 5) \cdot \arctg(\sqrt{14 \cdot e^{\sqrt{\pi x}} + \sin(2^{e \cdot \sqrt{\frac{\pi}{2}} \cdot x e^x})})\right)$$

$$f(x) = p((1985 - 1984) + 0 \cdot \operatorname{arctg}(\sqrt{14 \cdot e^{\sqrt{\pi}x}} + \sin(2e^{\sqrt{\frac{\pi}{2}} \cdot xe^x})))$$

Осталось заметить, что всё «огромное и страшное» просто умножается на ноль! Выбрасываем бóльшую часть функции и получаем следующее:

$$f(x) = p(1985 - 1984)$$

$$f(x) = p(1)$$

То есть $f(x)$ вообще не зависит от значения x , значение функции всегда равно $p(1)$, то есть $38 \cdot 1 + 4 = 42$. Таким образом, верное решение должно на любое значение, поданное на вход всегда выдавать число 42.

Возможные ошибки

- Сложности, разумеется, вызывала трудночитаемость формулы, но, преодолев данную трудность и заметив умножение на ноль, оставалось сделать лишь пару выводов на основе формулы.
- Общая проблема нескольких решений - использование посимвольного ввода *либо* ожидание ввода в конце работы программы *либо* заключение тела всей программы в бесконечный цикл. Данные проблемы встречались у нескольких участников и в разных задачах. Далее этот пункт упоминаться не будет.
- Считывание нужно проводить в переменную достаточного размера. Некоторые компиляторы вставляют проверки на границы допустимых значений типов переменных и выдают ошибку, если будет переполнение целочисленного типа.

Авторские решения

```

1  var                               1  input()
2      a: int64;                       2  print(42)
3
4  begin
5      Readln(a);
6      Writeln(42);
7  end.
```

2. Заговор

Условие задачи

В Кубитной империи всё не очень спокойно: офицер Комитета Компьютерного Контроля Владимир получил задание выявить всех членов заговора луддитов (людей, борющихся против механизации). Он блестяще выполнил свое задание, но, к сожалению, файл оказался повреждён: из него исчезли все переносы строк, но осталось все остальное форматирование. Изначально файл имел такой формат:

```

Имя1 Фамилия1
Имя2 Фамилия2
...
ИмяN ФамилияN
```

Некоторые из луддитов на самом деле агенты Комитета Компьютерного Контроля (после их фамилии стоит звёздочка):

```
Имя Фамилия*
```

Помогите Владимиру определить, сколько членов подпольной организации он выявил.

Формат входных данных

В единственной строке длины L ($3 \leq L \leq 10^8$) содержатся одна или несколько пар *Имя Фамилия*, но без переносов строк (после фамилии очередного члена заговора имя следующего начинается сразу следующим символом). Слова начинаются с большой буквы; сразу после фамилии (без дополнительного пробела), может следовать символ '*'. Допустимые символы в строке — символы латинского алфавита (a–z, A–Z), пробел и символ '*'.

Формат выходных данных

Целое число — количество упомянутых в повреждённом документе человек, разумеется, не включая агентов Комитета.

Категоризация

Азы работы со строковым типом данных, обработка длинных строк.

Задание считается лёгким, затруднения вызвала корректная обработка действительно длинных строк (порядка 10^7 символов).

Алгоритм решения

Очевидными являются следующие факты:

1. Количество членов заговора, включая агентов Комитета, равно количеству букв верхнего регистра (заглавных, A–Z), делённому на 2.
2. Количество агентов Комитета равно количеству символов '*' в исходной строке.

Отсюда следует решение: пройти посимвольно по строке, посчитать количество букв A–Z, разделить его на 2 и вычесть из полученного числа посчитанное количество символов '*'.

Возможные ошибки

- В некоторых языках программирования (или их диалектах, например Free Pascal) строки по умолчанию ограничены длиной в 255 символов. В PascalABC.NET и Python такого ограничения нет, во Free Pascal следовало использовать *ansistring*.
- Максимальные ограничения очень строги, они не допускают двойного прохода или неоптимального однократного прохода. Хорошей идеей было использовать файловый ввод, особенно в более медленных языках, вроде Python, использование ввода «с клавиатуры» приводило к выходу за ограничение по времени на последней, максимальной подгруппе тестов.

Авторские решения

```

1  var
2      s: string;
3      count: longint;
4      agents: longint;
5
6  begin
7      Readln(s);
8      foreach var c in s do
9          begin
10             if (c >= 'A') and (c <= 'Z') then
11                 count += 1
12             else if c = '*' then
13                 agents += 1;
14             end;
15             Writeln(count div 2 - agents);
16 end.

```

```

1  count, agents = 0, 0
2  with open('input.txt') as f:
3      for c in f.read():
4          if (c >= 'A') and (c <= 'Z'):
5              count += 1
6          elif c == '*':
7              agents += 1
8  print(count // 2 - agents)

```

3. Квантовые вычисления

Условие задачи

Квантовые компьютеры потребляют очень много электроэнергии, поэтому для Комитета Компьютерного Контроля Кубитной империи были построены несколько геотермальных электростанций. Но для работы электростанций требуется невероятно редкий (даже дефицитный!) материал Анобтаниум, поэтому мощность электростанций неодинакова и не всегда достаточна для нужд Империи. Возникает фундаментальная экономическая задача рационального распределения ресурсов.

Всего было построено N электростанций, для каждой известна её мощность, стоимость одного часа вырабатываемой электроэнергии и временной промежуток $[t_1, t_2]$, в течение которого электростанция готова обслуживать Комитет. В Комитете установлены стабилизаторы нагрузки, так что расход электроэнергии стабилен в течение каждого часа, а смена текущих поставщиков энергии (если таковая потребуется) происходит на рубеже часов. Гарантируется, что в любой момент времени найдётся не менее одной доступной электростанции. Антимонопольное

законодательство гарантирует: если энергосистема великой Кубитной империи может обеспечить работоспособность Кубитных Компьютеров, то будут задействованы все доступные электростанции. Высшее руководство Комитета Компьютерного Контроля хочет знать заранее количество часов простоя и итоговую стоимость поддержания работоспособности Комитета.

Примечание. При проверке на простой, нехватка мощности не более 10^{-6} (точность 6 знаков в дробной части числа) не считается значимой (расценивается как погрешность). В таком случае следует считать, что мощности хватает, простоя нет.

Формат входных данных

Первая строка: два целых числа N ($1 \leq N \leq 1000$), M ($1 \leq M \leq 86400$) и дробное R ($1 \leq R \leq 10^{300}$), где:

N — количество электростанций M — время работы компьютеров Комитета (в часах)

R — необходимая мощность для обеспечения бесперебойной работы Комитета

Следующие N строк: по четыре числа $t_{i_{\text{нач}}}$ $t_{i_{\text{кон}}}$ c p , где:

$t_{i_{\text{нач}}}$ и $t_{i_{\text{кон}}}$ — границы временного отрезка (в часах) доступности электростанции ($1 \leq t_{i_{\text{нач}}} \leq t_{i_{\text{кон}}} \leq m$)

c — её стоимость за час подачи электроэнергии ($1 \leq c \leq 10^9$)

p — её мощность ($1 \leq p \leq 10^{300}$, p — вещественное число).

Формат выходных данных

Два целых числа: количество часов, когда электроэнергии не хватит на обеспечение работы компьютеров, и оплаченную стоимость за вычетом тех часов, когда электроэнергии не будет хватать (когда мощность будет недостаточна, все компьютеры выключаются и электроэнергия не подаётся, таким образом оплачивать за простой не нужно).

Категоризация

Умение работать с вещественными числами, знание диапазонов допустимых значений типов данных, симуляция процессов на одномерном массиве.

Задание считается сложным, в связи с нестандартными ограничениями (10^{18} обычно является максимумом в олимпиадных заданиях, вещественными типами составители задач зачастую пренебрегают) и необходимостью написания простого и эффективного алгоритма, храня при этом данные максимально рационально.

Алгоритм решения

1. Считать данные N , M , R .
2. Подготовить 2 одномерных массива: мощностей и суммарной стоимости в каждый момент времени.
3. Для каждой электростанции дополнить значения массивов параметрами электростанции от начального времени до конечного.
4. Пройтись по обоим массивам одновременно, посчитать количество часов простоя и суммарную стоимость рабочих часов (не забывая сравнивать с заданной точностью).
5. Вывести ответ по формату.

Возможные ошибки

- Самое сложное – аккуратно посчитать ответ. От точности реализации работы с вещественными числами зависит судьба более половины баллов.
- Участники не имели опыта работы с погрешностями, в частности неверно интерпретировали примечание к условию, округляя до 6 знака после запятой непосредственно после ввода значений, а не на последнем шаге (при сравнении с R).
- Неэффективное хранение данных также часто встречалось в решениях участников.

Авторские решения

```

1 const
2     EPS: double = 1E-6;
3
4 var
5     n, m: longint;
6     R: double;
```

```

7     total: int64;
8     idle_time: int64;
9
10    begin
11        Readln(n, m, R);
12
13        var cost := new int64[m];
14        var power := new double[m];
15        for var i := 0 to n - 1 do
16            begin
17                var t_start, t_end: longint;
18                var c: int64;
19                var p: double;
20                Readln(t_start, t_end, c, p);
21                for var j := t_start - 1 to t_end - 1 do
22                    begin
23                        cost[j] += c;
24                        power[j] += p;
25                    end;
26            end;
27
28            { Подсчёт ответа }
29            for var i := 0 to m - 1 do
30                begin
31                    if power[i] + EPS < R then
32                        idle_time += 1
33                    else
34                        total += cost[i];
35                end;
36
37            Writeln(idle_time, ' ', total);
38    end.

```

```

1  import sys, codecs
2  sys.stdin = codecs.open("input.txt", "r", "utf-8")
3
4  values = input().split()
5  n, m, R = int(values[0]), int(values[1]), float(values[2])
6  cost, power = [0]*m, [0.0]*m
7  for i in range(n):
8      values = input().split()
9      t_start, t_end = int(values[0])-1, int(values[1])
10     c, p = int(values[2]), float(values[3])
11     for j in range(t_start, t_end):
12         cost[j] += c
13         power[j] += p
14
15     idle_time, total = 0, 0
16     for i in range(m):
17         if power[i] + 0.000001 < R:
18             idle_time += 1
19         else:
20             total += cost[i]
21
22     print(idle_time, total)

```

4. Элита имперских соревнований

Условие задачи

Все суровые программисты в великой Кубитной империи тренируются день и ночь для того, чтобы повысить свой рейтинг на Domestic Open Training Arena (имперской открытой тренировочной площадке). Лишь избранные

попадают в «зелёный» дивизион на CodeHorses, элитной олимпиадной площадке, причём по странному стечению обстоятельств вся элита происходит из столицы империи.

Василий Петрович уже не новичок, он сам тренирует команды. Он собирает команду из трёх талантливых молодых олимпиадников. Ему приходится выбирать тщательно, составляя сбалансированную команду. Учёные доказали, что для создания идеальной команды сумма рейтингов участников команды на тренировочной площадке должна делиться на определённое число k . Он уже отобрал четырёх кандидатов на три места. Помогите продвинутому тренеру выбрать трёх, чтобы составить идеальную команду. Если вдруг можно собрать несколько идеальных команд, то выведите любую. Можете быть уверены, идеальную команду всегда можно собрать, ведь теория вероятности в Кубитной империи весьма относительна.

Формат входных данных

Пять натуральных чисел a_1, a_2, a_3, a_4, k ($1 \leq a_1, a_2, a_3, a_4, k \leq 10^{16}$) — рейтинги четырёх кандидатов и так называемый *множитель идеальной команды*.

Формат выходных данных

Четыре натуральных числа i_1, i_2, i_3, s — номера выбранных участников в порядке возрастания ($1 \leq i_1 < i_2 < i_3 \leq 4$) и сумма их рейтингов ($s = n \cdot k, n \in \mathbb{N}$).

Категоризация

Условный оператор, основные арифметические операции.

Задание считается самым простым в списке задач данной олимпиады.

Алгоритм решения

В этой задаче было достаточно рассмотреть все 4 варианта взять 3 человек из 4 ($C_4^3 = \binom{4}{3} = 4$), и вывести ответ.

Возможные ошибки

- Неверный тип данных – более 60% неверных решений данного задания.
- Упущенные случаи для рассмотрения (например, только 2 из 4)

Авторские решения

```

1 var
2   a, b, c, d, k: int64;
3
4 begin
5   Readln(a, b, c, d, k);
6   if (a + b + c) mod k = 0 then
7     Writeln('1 2 3 ', (a + b + c))
8   else if (a + b + d) mod k = 0 then
9     Writeln('1 2 4 ', (a + b + d))
10  else if (b + c + d) mod k = 0 then
11    Writeln('2 3 4 ', (b + c + d))
12  else if (a + c + d) mod k = 0 then
13    Writeln('1 3 4 ', (a + c + d));
14 end.

```

```

1 a, b, c, d, k = map(int, input().split())
2 if (a+b+c) % k == 0:
3     print(1, 2, 3, (a+b+c))
4 elif (a+b+d) % k == 0:
5     print(1, 2, 4, (a+b+d))
6 elif (b+c+d) % k == 0:
7     print(2, 3, 4, (b+c+d))
8 elif (a+c+d) % k == 0:
9     print(1, 3, 4, (a+c+d))

```

5. Выгодный обмен

Условие задачи

Виктор Азинов решил в свободное время заняться торговлей на криптовалютной бирже. Виктор живёт на границе Байтляндии и Кубитной империи. В Байтляндии компьютеры полупроводниковые, и криптовалюты дорогие: один бетховен (вид криптовалюты) стоит n монет. В Кубитной империи все компьютеры квантовые, и они продают свои запасы бетховенов по m монет за штуку. Помогите решить задачу: рассчитайте, сколько раз Виктору, жителю Кубитной империи, понадобится пересечь границу, чтобы имея k монет стартового капитала получить капитал h монет.

Примечание: *капиталом* считаются только монеты, потому как Виктор отправляется в путешествие мечты в Ламповую республику, где бетховены считаются мыльным пузырьём и не подлежат обмену на монеты.

Формат входных данных

Четыре натуральных числа: n, m, k и h ($1 \leq n < m, k < h \leq 10^9$), где:

n — курс бетховенов в Кубитной империи k — стартовый капитал
 m — курс бетховенов в Байтляндии h — требуемая прибыль

Формат выходных данных

Единственное целое число – количество пересечений границ.

Категоризация

Умение понимать модель задачи и составлять алгоритм по текстовому описанию.
 Задание средней сложности, трудности возникали только с реализацией.

Алгоритм решения

Рассмотрим последовательно каждый шаг для симуляции процесса.

Сначала Виктор в Кубитной империи покупает бетховены ($b = \lfloor \frac{k}{n} \rfloor$ штук, то есть целочисленное деление k на n с округлением вниз). Затем он направляется в Байтляндию, где продаёт все имеющиеся бетховены и получает за них $b \cdot m$ монет, возвращается в Кубитную империю и цикл замыкается.

Реализуется данная задача на симуляцию также пошагово. Сначала вычисляется максимальное количество бетховенов, которое Виктор может купить в Кубитной империи, у него забирается определённое, кратное n число монет, добавляется кратное m число монет и количество переходов через границу увеличивается на 2. Данные операции выполняются пока количество монет не станет больше либо равно требуемому капиталу.

Возможные ошибки

- Неверное условие выхода из цикла и прочие мелкие ошибки

Авторские решения

```

1 var
2     m, n, k, h, buy_count, res: int64;
3
4 begin
5     Readln(n, m, k, h);
6     while k < h do
7         begin
8             buy_count := k div n;
9             k -= buy_count * n;
10            k += buy_count * m;
11            res += 1;
12        end;
13        res *= 2;
14        Writeln(res);
15 end.
```

```

1 n, m, k, h = map(int, input().split())
2 res = 0
3 while k < h:
```

```

4     buy = k // n
5     k -= buy * n
6     k += buy * m
7     res += 1
8
9     print(res*2)

```

6. Восстановление порядка

Условие задачи

После криптовалютных махинаций Виктора в Байтляндии с экономикой стало всё плохо. В качестве чрезвычайной меры правители Байтляндии (все восемь министров), решили запретить продажу бетховенов в столице. Город представляет собой таблицу из N строк и M столбцов, каждая клетка — дом, в котором продается криптовалюта по заданной цене (или не продаётся, не все жители доверяли Виктору). Правительство каждый день скупает все бетховены на строке с номером i и столбце с номером j . Определите, в какой клетке после антикризисных мер выгоднее всего купить бетховены. Если после действий авторитарного правительства купить криптовалюту оказывается невозможным, выведите -1 .

Формат входных данных

Первая строка: три натуральных числа N , M и K , где N , M — количество строк и столбцов в таблице ($1 \leq N, M \leq 3 \cdot 10^3$), K — количество дней с правительственными закупками ($1 \leq K \leq 366$).

Следующие N строк: по M целых чисел c_{ij} ($1 \leq c_{ij} \leq 10^{18}$ или $c_{ij} = -1$), курс бетховена в очередной клетке таблицы или -1 , если бетховены здесь не продаются.

Следующие K строк: по два натуральных числа i и j , где i, j — скупаемая строчка и столбец соответственно ($1 \leq i \leq N, 1 \leq j \leq M$)

Формат выходных данных

Три натуральных числа – координаты клетки (строка, столбец) и курс бетховена в клетке с самым выгодным курсом, если покупка возможна. В противном случае – единственное число -1 .

Категоризация

Двумерные массивы данных, простая симуляция процессов.

Задание считается трудным, в связи с необходимостью обработки случая -1 в клетках таблицы и аккуратной работы с индексами клеток.

Алгоритм решения

Ещё одна задача, алгоритмизация которой не составляет особого труда (фактически, весь алгоритм дан в условии задания), но требуется оптимально и быстро реализовать симуляцию.

Возможные ошибки

- Участники пропускали проверку на -1 в клетках таблицы при подсчёте ответа.
- Ученики выходили за границы массива, путая граничные значения вложенных циклов.
- Неоптимально пересчитывать минимум после каждой операции, достаточно это сделать один раз, после всех операций скупки криптовалюты.

Авторские решения

```

1     var
2         n, m, k, res_i, res_j: integer;
3         matrix: array[,] of int64;
4         res: int64;
5
6     begin
7         Readln(n, m, k);
8         matrix := new int64[n, m];
9

```



```

10 { В PascalABC.NET динамические двумерные массивы индексируются с 0 }
11 n := 1;
12 m := 1;
13
14 { Считывание }
15 for var i := 0 to n do
16     for var j := 0 to m do
17         Read(matrix[i, j]);
18
19 { Обработка }
20 for var t := 1 to k do
21     begin
22         var i, j: integer;
23         Readln(i, j);
24         i -= 1;
25         j -= 1;
26         for var idx := 0 to n do
27             matrix[idx, j] := -1;
28         for var idx := 0 to m do
29             matrix[i, idx] := -1;
30     end;
31
32 { Подсчёт ответа }
33 res := int64.MaxValue;
34 res_i := -1;
35 res_j := -1;
36 for var i := 0 to n do
37     for var j := 0 to m do
38         begin
39             var value := matrix[i, j];
40             if (value <> -1) and (value < res) then
41                 begin
42                     res_i := i;
43                     res_j := j;
44                     res := value;
45                 end;
46         end;
47
48 if res = int64.MaxValue then
49     Writeln(-1)
50 else
51     Writeln((res_i + 1), ' ', (res_j + 1), ' ', res);
52 end.

```

```

1 import sys, codecs
2 sys.stdin = codecs.open("input.txt", "r", "utf-8")
3
4 n, m, k = map(int, input().split())
5 matrix = [None]*n
6 for i in range(n):
7     matrix[i] = list(map(int, input().split()))
8
9 for t in range(k):
10     i, j = map(int, input().split())
11     i, j = i-1, j-1
12     for idx in range(n):
13         matrix[idx][j] = -1
14     for idx in range(m):
15         matrix[i][idx] = -1
16
17 res, res_i, res_j = 10**36, -1, -1
18 for i in range(n):
19     for j in range(m):

```

```
20     value = matrix[i][j]
21     if value == -1:
22         continue
23     if value >= res:
24         continue
25     res, res_i, res_j = value, i, j
26
27 if res_i < 0:
28     print(-1)
29 else:
30     print((res_i+1), (res_j+1), res)
```