

Московская олимпиада школьников по информатике 2025-2026 6-7 класс (второй отбор)

4 янв 2026 г., 12:00 — 15 янв 2026 г., 23:59

100 баллов

Карусель

Не начинайте решение задач, пока вы не включили запись экрана с видео участника, см. регламент олимпиады, без корректной записи работа не будет засчитана

В парке аттракционов есть карусель, которая вмещает в себя k человек. Одно катание на карусели длится t минут. Сейчас h часов m минут, Петя n -й в очереди на аттракцион. Ему интересно, успеет ли он прокатиться до того, как парк закроется в h_2 часов m_2 минут. А если успеет, то сколько времени ему придется ждать.

Карусель работает с текущего момента до времени закрытия включительно.

Входные данные

В первой строке находятся три числа k, t, n , ($1 \leq k, t, n \leq 10^9$) — вместимость карусели, продолжительность катания и номер Пети в очереди соответственно.

Во второй строке находятся два числа h, m — текущее время.

В третьей строке находятся два числа h_2, m_2 количество часов и минут время закрытия парка ($0 \leq h, h_2 < 24$), ($0 \leq m, m_2 < 60$).

Числа в строках разделены одним пробелом.

Гарантируется, что текущий момент времени строго меньше времени закрытия.

Выходные данные

Если Петя не дождется своей очереди, выведите -1.

Иначе выведите сколько времени в минутах Пете придется ждать своей очереди.

Примеры

```
4 2 5
10 5
10 10
```

```
2
```

```
4 4 5
10 5
10 10
```

```
-1
```

Ограничения

Время выполнения: 1 секунда

Память: 256 МВ

Код

C++

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7  ....
8  .... int k, t, n, h1, m1, h2, m2;
9  .... cin >> k >> t >> n;
10 .... cin >> h1 >> m1;
11 .... cin >> h2 >> m2;
12 .... int m = (h2 * 60 + m2) - (h1 * 60 + m1) + 1;
13 .... if (m / t >= n / k + (n % k != 0))
14 ....     cout << n / k * t - (n % k == 0) * t;
15 .... else
16 ....     cout << -1;
17 .... return 0;
18 }
19
```

100 баллов

Затор на Транссибе

Не начинайте решение задач, пока вы не включили запись экрана с видео участника, см. регламент олимпиады, без корректной записи работа не будет засчитана

На одном из участков Транссибирской магистрали возникло временное ограничение движения. В результате n поездов, следующих один за другим в сторону Владивостока, оказались в ситуации «цепочки».

Каждый i -й поезд приближается к участку со своей изначальной скоростью a_i . Однако из-за специфики движения по колее возникает физическое ограничение: ни один поезд не может двигаться быстрее, чем поезд, идущий непосредственно перед ним.

Если скорость поезда a_i больше скорости впереди идущего поезда, то i -й поезд вынужден замедлиться до его скорости. Поезда прибывают к переезду в порядке нумерации от 1 до n .

Ваша задача — определить установившуюся скорость для каждого из n составов.

Входные данные

В первой строке вводится целое число n ($1 \leq n \leq 10^5$) — количество поездов. Во второй строке через пробел вводятся n целых чисел a_i ($1 \leq a_i \leq 10^{18}$) — исходные скорости.

Выходные данные

Выведите n чисел — итоговую скорость каждого поезда после вынужденного замедления.

Примеры

```
3
100 80 100
```

```
100 80 80
```

```
5
100 80 60 100 80
```

```
100 80 60 60 60
```

Код

C++

```

1  #include <bits/stdc++.h>
2
3  #define ll long long
4  template<typename T>
5  bool smin(T& a, const T& b) {
6  .... if (b < a) {
7  ....     a = b;
8  ....     return true;
9  .... }
10 .... return false;
11 }
12
13 template<typename T>
14 bool smax(T& a, const T& b) {
15 .... if (a < b) {
16 ....     a = b;
17 ....     return true;
18 .... }
19 .... return false;
20 }
21
22 const int INF = 0x3f3f3f3f;
23 const ll LINF = (ll)2e18;
24 const int N = 1000100;
25
26 ll a[N];
27
28 void run() {
29 .... int n;
30 .... scanf("%d", &n);
31
32 .... for (int i = 0; i < n; ++i) {
33 ....     scanf("%lld", &a[i]);
34 ....     if (i != 0) {
35 ....         smin(a[i], a[i - 1]);
36 ....     }
37 .... }
38
39 .... for (int i = 0; i < n; ++i) {
40 ....     printf("%lld ", a[i]);
41 .... }
42 .... printf("\n");
43 }
44
45 int main(void) {
46 .... run();
47
48 .... return 0;
49 }
50

```

100 баллов

Будни авиадиспетчера

Не начинайте решение задач, пока вы не включили запись экрана с видео участника, см. регламент олимпиады, без корректной записи работа не будет засчитана

Максим работает авиадиспетчером в аэропорту с единственной взлетно-посадочной полосой (ВПП). Сегодня ему предстоит посадить n самолетов, и ваша задача — рассчитать время завершения этого процесса.

Правила посадки:

1. Продолжительность: Процесс посадки одного самолета занимает ровно b минут. В это время ВПП считается занятой.
2. Очередь и круги: Если самолет прибывает к аэропорту, а ВПП занята, он отправляется на дополнительный круг. Один круг занимает f минут. После завершения круга самолет снова оказывается у ВПП и проверяет её готовность.
3. Свободная полоса: Если в момент прибытия (или возвращения с круга) ВПП свободна, самолет немедленно приступает к посадке.
4. Конфликты: Если несколько самолетов оказываются у свободной ВПП одновременно, один из них (любой) начинает посадку, а остальные немедленно отправляются на дополнительный круг длительностью f минут.

Задание: Определите момент времени, когда последний самолет завершит свою посадку.

Входные данные

Первая строка входных данных содержит три целых числа n , b и f ($1 \leq n \leq 1000$, $1 \leq b, f \leq 10^9$).

Вторая строка входных данных содержит n целых чисел t_i ($0 \leq t_i \leq 10^9$) — времена первоначального прибытия самолетов (в произвольном порядке).

Выходные данные

Выведите одно число — время, в которое последний самолет закончит посадку.

Примеры

```
10 10 12
13 0 1 10 20 20 2 1 10 20
```

```
120
```

Ограничения

Время выполнения: 1 секунда

Память: 256 МВ

Код

C++

```
1 #include <iostream>
2 #include <map>
3 #include <set>
4 using namespace std;
5 #define ll long long
6 int main() {
7     ll n, b, f; cin >> n >> b >> f;
8     multiset<ll> s;
9     for (int i = 0; i < n; i++) {
10         ll x; cin >> x;
11         s.insert(x);
12     }
13     ll cur = 0;
14     while (s.size() != 0) {
15         ll x = *(s.begin());
16         s.erase(s.begin());
17         if (x >= cur) cur = x + b;
18         else {
19             ll t = (cur - x) / f;
20             if ((cur - x) % f != 0) t++;
21             s.insert(t * f + x);
22         }
23     }
24     cout << cur;
25 }
```

№ 4

100 баллов

Готовим к ЕГЭ по информатике старших братьев

Не начинайте решение задач, пока вы не включили запись экрана с видео участника, см. регламент олимпиады, без корректной записи работа не будет засчитана

В ЕГЭ по информатике может встретиться следующая задача. Дана последовательность прописных латинских букв и цифр. Определить максимальную длину подряд идущих цифр, сумма которых делится на k .

Входные данные

Первая строка входных данных содержит одно целое положительное число k , на которое должна делиться сумма последовательности из цифр. ($1 \leq k \leq 10^3$).

Вторая строка содержит непустую строку s . Строка состоит из прописных латинских букв и цифр. Длина строки s не превышает 10^6 .

Выходные данные

Выведите одно целое число — длину искомой последовательности.

Система оценивания

В этой задаче три группы тестов. В первой группе длина строки не превосходят 100, во второй группе — 1000, в третьей группе нет дополнительных ограничений. Баллы за группу можно получить только при прохождении всех тестов группы и всех тестов предыдущих групп.

Примеры

```
3
ab99c112111d
```

```
5
```

```
3
ab7c11d
```

```
0
```

```
1
x999ууу0123456789
```

```
10
```

Ограничения

Время выполнения: 3 секунды

Память: 256 МВ

Код

C++

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <cmath>
5  #include <map>
6  #include <set>
7  #include <unordered_map>
8  #include <queue>
9
10 using namespace std;
11
12 #define int long long
13
14 bool isdigit(char c){
15     .... return c >= '0' && c <= '9';
16 }
17
18 signed main()
19 {
20     .... ios::sync_with_stdio(0);
21     .... cout.tie(0);
22     .... cin.tie(0);
23     ....
24     .... int k;
25     .... cin >> k;
26     ....
27     .... string s;cin >> s;
28     .... int n = s.size();
29     ....
30     .... int ans = 0;
31     .... for(int i = 0; i < n; i++){
32         .... if(!isdigit(s[i]))continue;
33         ....
34         .... int j = i;
35         .... int sum = s[i] - '0';
36         .... unordered_map<int, int> mp;
37         .... mp[0] = i - 1;
38         ....
39         .... if(sum % k == 0){
40             .... ans = max(ans, 1ll);
41         .... }
42         .... else{
43             .... mp[sum%k] = i;
44         .... }
45         ....
46         .... while(j + 1 < n && isdigit(s[j+1])){
47             .... j++;
48             .... sum += s[j] - '0';
49             .... if(mp.count(sum % k))ans = max(ans, j - mp[sum%k]);
50             .... else mp[sum%k] = j;
51         .... }
52         ....
53         .... i = j;
54     .... }
55     ....
56     .... cout << ans;
57     .... return 0;
58 }

```

100 баллов

Восстановление исходного массива

Не начинайте решение задач, пока вы не включили запись экрана с видео участника, см. регламент олимпиады, без корректной записи работа не будет засчитана

Представьте, что у вас был массив из N натуральных чисел a_1, a_2, \dots, a_N . На его основе вы построили таблицу $N \times N$ по следующим правилам:

1. В ячейках на главной диагонали (где номер строки i совпадает с номером столбца j) всегда записывается 0.
2. В остальных ячейках (где $i \neq j$) записывается максимальное из двух чисел: a_i и a_j .

Вам дана уже готовая таблица. Требуется «расшифровать» её и найти исходный массив a_1, \dots, a_N . Если такая таблица не могла быть получена по этим правилам, выведите -1 .

Входные данные

В первой строке находится число N ($1 \leq N \leq 500$) — размер таблицы. В последующих N строках находятся по N целых чисел (от 0 до 1000) — сама таблица.

Входные данные

Если решение существует, выведите N чисел через пробел. Если решений несколько, выведите любое. Если решения нет, выведите -1 .

Примеры

```
3
0 4 6
4 0 6
6 6 0
```

```
4 4 6
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код

C++

```

1  #include <iostream>
2  #include <algorithm>
3  #include <cmath>
4  #include <string>
5  #include <vector>
6  using namespace std;
7  typedef long long ll;
8  const int MAX_N = 2e3+6;
9  const double INF = 1e9+1;
10 int main() {
11     int n; cin >> n;
12     vector<vector<int>> matrix(n, vector<int>(n));
13     for (int i = 0; i < n; i++) {
14         for (int j = 0; j < n; j++) {
15             cin >> matrix[i][j];
16         }
17     }
18     vector<int> a(n);
19     for (int i = 0; i < n; i++) {
20         int mn = 100001;
21         for (int j = 0; j < n; j++) {
22             if (j == i) continue;
23             mn = min(mn, matrix[i][j]);
24             if (matrix[i][j] != matrix[j][i]) {
25                 cout << -1 << endl;
26                 return 0;
27             }
28         }
29         if (mn == 100001 || mn == 0) mn = 1;
30         a[i] = mn;
31         if (matrix[i][i] != 0) {
32             cout << -1 << endl;
33             return 0;
34         }
35     }
36     for (int i = 0; i < n; i++) {
37         for (int j = 0; j < n; j++) {
38             if (i == j) continue;
39             if (matrix[i][j] != max(a[i], a[j])) {
40                 cout << -1 << endl;
41                 return 0;
42             }
43         }
44     }
45     for (int x : a) {
46         cout << x << ' ';
47     }
48 }

```