Московский авиационный институт (национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Журнал по исследовательской практике (индивидуальный план)

Студенты	Группа
Кузьмичев Александр Николаевич	М8О-306Б-18
Кудинов Сергей Владимирович	М8О-306Б-18
Орозбакиев Эмиль Даниярович	М8О-306Б-18
Команда:	MAI #38: Kudinov, Kuzmichev, Orozbakiev

Москва, 2021

Сводная таблица за осень 2020

Дата	Название	Время	Место проведения	Решенные задачи
27.09.2020	Grand Prix of Eurasia	11:00-16:00	Дистанционно	1, 2, 7
04.10.2020	RuCode Championship 2020 Div C	11:30-16:30	Дистанционно	A, B, C, D, E
11.10.2020	Grand Prix of Korea	11:00-16:00	Дистанционно	B, D, G, K
01.11.2020	Grand Prix of Siberia	11:00-16:00	Дистанционно	1, 10, 11
08.11.2020	Grand Prix of Weihai	11:00-16:00	Дистанционно	B, I, L, M, N, O, P
15.11.2020	MRC-2020. Main Contest	11:00-16:00	Дистанционно	A, B, C, D
22.11.2020	Regional 1. Northwestern Russia 2020	11:00-16:00	Дистанционно	A, B, I
29.11.2020	Grand Prix of NorthBeach	11:00-16:00	Дистанционно	K, L, M

Явка на контесты

Дата	Название	Присутствующие
27.09.2020	Grand Prix of Eurasia	Кузьмичев, Кудинов, Орозбакиев
04.10.2020	RuCode Championship 2020 Div C	Кузьмичев, Кудинов, Орозбакиев
11.10.2020	Grand Prix of Korea	Кузьмичев, Кудинов, Орозбакиев
01.11.2020	Grand Prix of Siberia	Кузьмичев, Кудинов, Орозбакиев
08.11.2020	Grand Prix of Weihai	Кузьмичев, Кудинов, Орозбакиев
15.11.2020	MRC-2020. Main Contest	Кузьмичев, Кудинов, Орозбакиев
22.11.2020	Regional 1. Northwestern Russia 2020	Кузьмичев, Кудинов, Орозбакиев
29.11.2020	Grand Prix of NorthBeach	Кузьмичев, Кудинов, Орозбакиев

Решения задач

Grand Prix of Eurasia 27.09.2020

Run ID	Time	Size	Problem	Language	Result	Failed test	View source	View report
237	3:25:15	1236	7	g++0x	Wrong answer	2	View	View
222	3:13:05	1251	7	g++0x	Wrong answer	2	View	View
219	3:12:05	1205	7	g++0x	Wrong answer	2	View	View
216	3:10:28	1227	7	g++0x	Run-time error	1	View	View
201	2:55:40	1093	7	g++0x	Wrong answer	2	View	View
190	2:47:33	1747	7	g++0x	Wrong answer	2	View	View
188	2:45:30	688	2	g++	Wrong answer	2	View	View
185	2:44:41	1671	7	g++0x	Wrong answer	2	View	View
184	2:44:10	681	2	g++0x	Presentation error	2	View	View
158	2:17:05	1649	7	g++0x	Wrong answer	2	View	View
153	2:14:20	1602	7	g++0x	Wrong answer	2	View	View
144	2:06:54	1342	7	g++0x	Wrong answer	3	View	View
140	2:06:01	1342	7	g++	Compilation error	N/A	View	View
81	1:03:09	359	1	g++	OK	N/A	View	View
49	0:44:29	345	1	g++	Wrong answer	2	View	View
46	0:43:22	345	1	g++	Wrong answer	1	View	View

Решение 1

```cpp
#include <iostream>

using namespace std;

int main() {
    int T;
    cin >> T;
    for(int i = 0; i < T; i++) {
        long long N, M, R, C, S;
        cin >> N >> M >> R >> C >> S;
        if (((C - S > 1) && (R + S < N)) || ((R - S > 1) && (C + S < M))) {
            cout << "Barsik"<< endl;
        } else cout << "Tuzik" << endl;
    }
    return 0;
}
```

Решение 2 (Не засчитано)

```cpp
#include <iostream>

using namespace std;
```

```
4
5  int main() {
6      double t, p, q, a, b;
7      cin >> t;
8      for (int i = 0; i < t; i++){
9          double v1, v2, v1Kodera, v2Kodera;
10
11         cin >> p >> q >> a >> b;
12
13         v1=1;
14         v2=q / p;
15         v1Kodera = a * v1;
16         v2Kodera = v2 / b;
17         p=100-p;
18         q=100-q;
19         int j =0;
20         while(true){
21             if((p/v1)<=(q/v2)){
22                 cout << j << endl;
23                 break;
24             } else {
25                 j++;
26                 v1+=v1Kodera;
27                 v2+=v2Kodera;
28             }
29             if (v2 > 100){
30                 cout << -1<< endl;
31                 break;
32             }
33         }
34     }
35 }
```

Решение 7 (Не засчитано)

```
1  #include <vector>
2
3  #include <iostream>
4  #include <queue>
5  #include <vector>
6  struct TEdge {
7      long long to;
8      long long from;
9  };
10 struct TData {
11     std::vector<long long> parent;
12     std::vector<bool> visited;
13     std::vector<std::vector<long long>> graph;
14     long long n,m;
15 };
```

```
16
17 int main() {
18     long long tests;
19
20     std::cin >> tests;
21     long long n;
22     int hasRoad;
23     long long v;
24     std::vector<TEdge> edges;
25     std::vector<TEdge> edges1;
26
27     std::vector<bool> isVisited;
28     std::vector<long long> ans;
29     for (auto i = 0; i < tests; ++ i) {
30         std::cin >> n;
31         ans.push_back(0);
32         isVisited.resize(n,false);
33         for (auto j = 0; j < n; ++j) {
34             v = 0;
35             for (auto k = 0; k <n; ++ k) {
36                 std::cin >> hasRoad;
37                     if (hasRoad==1 && isVisited[k]==false) {
38                         v++;
39                         isVisited[k] = true;
40                     }
41                 }
42                 v--;
43                 if (v>0) {
44                     ans[ans.size()-1]+=(v);
45                 }
46         }
47     }
48     for (int i = 0 ; i < ans.size(); ++i) {
49         std::cout << ans[i] << std::endl;
50
51     }
52 }
```

Grand Prix of Korea 11.10.2020

Problem status summary

Short name	Long name	Input file name	Output file name	Time limit	Language TL's	Memory limit	Stack limit	Source limit	Submits left	Status	Failed test	Run ID
A	Advertisement Matching	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
B	Bombs in my Deck	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	1999	OK		259
C	Economic One-Way Roads	**input.txt** or *standard input*	**output.txt** or *standard output*	15000 ms		1024 M	64 M	65536 bytes	2000			
D	Fix Wiring	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
E	Min-hashing	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
F	Square, Not Rectangle	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
G	LCS 8	**input.txt** or *standard input*	**output.txt** or *standard output*	4000 ms		1024 M	64 M	65536 bytes	2000			
H	Mock Competition Marketing	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
I	Query on a Tree 17	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
J	Remote Control	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
K	Sewing Graph	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
L	Steel Slicing 2	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			

My Submissions

#	When	Who	Problem	Lang	Verdict	Time	Memory
95707796	6 days	Olimp_8F: enstein225, pokemo4ik, poisoned_monkey	D - Fix Wiring	GNU C++14	**Happy New Year!**	31 ms	200 KB
95707599	6 days	Olimp_8F: enstein225, pokemo4ik, poisoned_monkey	D - Fix Wiring	GNU C++14	Wrong answer on test 2	31 ms	0 KB
95629453	5 days	Olimp_8F: enstein225, pokemo4ik, poisoned_monkey	G - Mock Competition Marketing	GNU C++14	**Happy New Year!**	46 ms	1100 KB
95625079	4 days	Olimp_8F: enstein225, pokemo4ik, poisoned_monkey	K - Square, Not Rectangle	GNU C++14	**Happy New Year!**	109 ms	3100 KB

Решение B

```cpp
#include <iomanip>
#include <iostream>
#include <vector>
struct Tdata {
    size_t sum;
    size_t lastElem;
};
int main() {

    int a,b,c;
    std::cout << std::fixed;

    std::cout.precision(8);
    std::cin >> a >> b >> c;
    double prob = 1;
    while (c>0 && b>0 && a>0) {
        prob = prob*((double)b/(double)a);
        b--;
        a--;
        c = c -5;
    }
    if (b==0 && c>0) {

```

```
        std::cout << 1;
    } else {

        std::cout << 1.-prob;
    }
    return 0;
}
```

Решение D

```
#include <bits/stdc++.h>


int main() {

    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    long long n;
    std::cin >> n;
    long long m = n*(n-1)/2;
    std::vector<long long> costs;
    long long cost;
    for (auto i = 0; i < m; ++i) {
        std::cin >> cost;
        costs.push_back(cost);
    }
    std::sort(costs.begin(),costs.end());
    long long minimum = 0;
    for (auto i = 0; i < n-1; ++i) {
        minimum+=costs[i];
    }
    long long maximum = 0;
    for (auto i = 0; i <n-1; ++i) {
        maximum+=costs[(i+1)*(i)/2];
    }
    std::cout << minimum << " " << maximum;
    return 0;
}
```

Решение G

```
#include <bits/stdc++.h>


bool sortbysec(const std::pair<size_t,size_t> &a,
               const std::pair<size_t,size_t> &b)
{
    return (a.second < b.second);
}


```

```
11 int main() {
12     std::ios_base::sync_with_stdio(false);
13     std::cin.tie(NULL);
14     size_t n, k;
15     std::cin >> n >> k;
16     size_t cost;
17     std::vector<std::pair<size_t, size_t>> ad_types;
18     std::vector<std::pair<size_t, size_t>> sorted_types;
19 
20     for (auto i = 0; i < 6 ; ++i) {
21         std::cin >> cost;
22         ad_types.push_back(std::make_pair(i,cost));
23         sorted_types.push_back(std::make_pair(i,cost));
24     }
25     std::sort(sorted_types.begin(),sorted_types.end(),sortbysec);
26     size_t types = 0;
27     std::vector<size_t> all_ads;
28     size_t now_type;
29     for (auto i = 0; i < n; ++i) {
30         std::cin >> now_type;
31         all_ads.push_back(now_type);
32     }
33     size_t max_ads = 0;
34     size_t now_ads;
35     std::set<size_t> type_allowed;
36     size_t k_now;
37     while (types<6) {
38         type_allowed.insert(sorted_types[types].first);
39         // for (auto it = sorted_types.begin(); it!= sorted_types.end(); it++) {
40         // std::cout << (*it).first;
41         // }
42         k_now = k;
43         now_ads = 0;
44         for (auto i = 0; i < n; ++i) {
45             if (type_allowed.count(all_ads[i]-1)==1 && ad_types[all_ads[i]-1].second<=
                    k_now) {
46 
47                 k_now-=ad_types[all_ads[i]-1].second;
48                 now_ads++;
49             }
50         }
51         if (now_ads>max_ads) {
52             max_ads = now_ads;
53         }
54         types++;
55     }
56     std::cout << max_ads;
57     return 0;
58 }
```

Решение K

```cpp
#include <bits/stdc++.h>

bool checkSquare(std::vector<size_t>& vect, size_t m) {
    size_t m1 = m;
    size_t ind = 0;
    while (ind < vect.size()) {
        if (vect[ind]>=m) {
            m1--;
        }
        else {
            m1 = m;
        }
        if (m1==0) {
            return true;
        }
        ind++;
    }
    return false;
}
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    size_t n;
    std::vector<size_t> histogramm;
    size_t h;
    std::cin >> n;
    for (auto i = 0; i <n; ++i) {
        std::cin >> h;
        histogramm.push_back(h);
    }
    size_t l = 1;
    size_t r = 300000;
    size_t m = 0;
    while (r-l>1) {
        m = l+(r-l)/2;
        if (checkSquare(histogramm, m)==true) {
            l = m;
        } else {
            r = m;
        }
    }
    if (checkSquare(histogramm,r)==true) {
        std::cout << r;
    } else if (checkSquare(histogramm,m)==true) {
        std::cout << m;
    } else if (checkSquare(histogramm,l)==true) {
        std::cout << l;
    }
```

```
    return 0;
}
```

Grand Prix of Siberia 01.11.2020

Problem status summary

Short name	Long name	Input file name	Output file name	Time limit	Language TL's	Memory limit	Stack limit	Source limit	Submits left	Status	Failed test	Run ID
1	Complicated Document	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		256 M	256 M	65536 bytes	1996	OK		256
3	Find the radio operator	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	2000			
4	Morse Code	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	2000			
5	Woodcut	**input.txt** or *standard input*	**output.txt** or *standard output*	5000 ms		256 M	256 M	65536 bytes	2000			
7	Scheduler	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	2000			
9	Crusing Blow	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		256 M	256 M	65536 bytes	2000			
10	Antiplaglarism	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		256 M	256 M	65536 bytes	1996	OK		377
11	Files List	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		256 M	256 M	65536 bytes	1999	OK		396
12	Hive	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		256 M	256 M	65536 bytes	2000			
13	Side Effects	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		256 M	256 M	65536 bytes	2000			

This is **ejudge** contest administration system, version 2.3.27 #245, compiled Sun Feb 23 04:22:43 2020.

Решение 1

```
#include <bits/stdc++.h>
char lastC = ' ';

 bool isNeedChar(char c) {
    return (c=='-' || c==':');
 }
 void print(char c) {
    if (!(lastC==' ' && c==' ')) {
        std::cout << c;
        lastC = c;
    }
 }
int main() {

   std::ios_base::sync_with_stdio(false);
   std::cin.tie(0);
   size_t n;
   std::cin >> n;
   std::string a;
   std::getline(std::cin, a);
        std::vector<std::string> ans;
   for (auto i = 0; i < n; ++i) {
       std::string line1;
       std::getline (std::cin,line1);
       ans.push_back(line1);
   }
   for (auto i = 0; i < n; ++i) {
       std::string line1 = ans[i];
       line1.push_back('\n');
       std::cout << line1[0];
       lastC = line1[0];
       if (line1.size()==1) {
```

```
            std::cout << std::endl;
            continue;
        }

        for (int j = 1; j < line1.size(); ++j) {
           if (isNeedChar(line1[j]) && isNeedChar(lastC)) {
               std::cout << ' ' << line1[j];
               lastC = line1[j];
           } else if (isNeedChar(line1[j]) && lastC!=' ') {
               std::cout << ' ' << line1[j];
               lastC = line1[j];
           } else if (!isNeedChar(line1[j]) && isNeedChar(lastC) && line1[j]!=' ' &&
               line1[j]!='\n') {
               std::cout << ' ' << line1[j];
               lastC = line1[j];
           } else {
               std::cout << line1[j];
               lastC = line1[j];
           }
        }
    }


    return 0;
}
```

Решение 10

```
#include <bits/stdc++.h>
    size_t n;


void route(std::vector<std::vector<char>>& arr) {
     for (int x = 0; x < n / 2; x++) {

      for (int y = x; y < n - x - 1; y++) {

            char temp = arr[x][y];
            arr[x][y] = arr[y][n - 1 - x];
            arr[y][n - 1 - x] = arr[n - 1 - x][n - 1 - y];
            arr[n - 1 - x][n - 1 - y] = arr[n - 1 - y][x];
            arr[n - 1 - y][x] = temp;
      }
   }
}
void Horizontal(std::vector<std::vector<char>>& arr) {
    for (int i = 0; i < n / 2 ; ++i)
   {
       for (int j = 0; j < n; ++j)
       {
```

```
            char tmp = arr[i][j];
            arr[i][j] = arr[n - i - 1][j];
            arr[n - i - 1][j] = tmp;
        }
    }
}
void Vertical(std::vector<std::vector<char>>& arr) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n / 2; j++) {
            char tmp = arr[i][j];
            arr[i][j] = arr[i][n - j-1];
            arr[i][n - j-1] = tmp;
        }
    }
}
bool check(std::vector<std::vector<char>>& arr1, std::vector<std::vector<char>>& arr2)
     {
    bool flag = true;
    for (int i = 0 ; i < n; ++i) {
        for (int j = 0; j<n; ++j) {
            if (arr1[i][j]!=arr2[i][j]) {
                return false;
            }
        }
    }
    return true;
}
void print(std::vector<std::vector<char>>& arr1) {
// for (int i = 0 ; i < n; ++i) {
// for (int j = 0; j<n; ++j) {
// std::cout << arr1[i][j];
// }
// std::cout << std::endl;
// }
// std:: cout << std::endl;
}
int main() {

    std::ios_base::sync_with_stdio(false);
    std::cin.tie(0);
    std::cin >> n;
    std::vector<std::vector<char>> arr(n,std::vector<char>(n,'*'));
        std::vector<std::vector<char>> arr1(n,std::vector<char>(n,'*'));
        std::string s;
    std::getline(std::cin, s);
    for (int i = 0; i < n; ++i) {
            std::getline(std::cin, s);

        for (int j = 0; j < n; ++j) {
```

```
71
72              arr[i][j] = s[j];
73
74          }
75      }
76      std::getline(std::cin,s);
77        for (int i = 0; i < n; ++i) {
78              std::getline(std::cin, s);
79
80          for (int j = 0; j < n; ++j) {
81
82              arr1[i][j] = s[j];
83
84          }
85      }
86      print(arr1);
87      for (int i = 0; i < 4; ++i) {
88          route(arr);
89          print(arr);
90          if (check(arr,arr1)) {
91              std::cout << "YES" << std::endl;
92              return 0;
93          }
94
95      }
96      Vertical(arr);
97      for (int i = 0; i < 4; ++i) {
98          route(arr);
99                  print(arr);
100
101         if (check(arr,arr1)) {
102             std::cout << "YES" << std::endl;
103             return 0;
104         }
105
106     }
107     Vertical(arr);
108     Horizontal(arr);
109     for (int i = 0; i < 4; ++i) {
110         route(arr);
111         print(arr);
112
113         if (check(arr,arr1)) {
114             std::cout << "YES" << std::endl;
115             return 0;
116         }
117
118     }
119     Horizontal(arr);
```

```
    std::cout << "NO" << std::endl;
    return 0;
}
```

Решение 11

```
#include <bits/stdc++.h>
    size_t n;


 void route(std::vector<std::vector<char>>& arr) {
      for (int x = 0; x < n / 2; x++) {

       for (int y = x; y < n - x - 1; y++) {

             char temp = arr[x][y];
             arr[x][y] = arr[y][n - 1 - x];
             arr[y][n - 1 - x] = arr[n - 1 - x][n - 1 - y];
             arr[n - 1 - x][n - 1 - y] = arr[n - 1 - y][x];
             arr[n - 1 - y][x] = temp;
       }
   }
 }
 void Horizontal(std::vector<std::vector<char>>& arr) {
    for (int i = 0; i < n / 2 ; ++i)
   {
       for (int j = 0; j < n; ++j)
       {
           char tmp = arr[i][j];
           arr[i][j] = arr[n - i - 1][j];
           arr[n - i - 1][j] = tmp;
       }
   }
 }
 void Vertical(std::vector<std::vector<char>>& arr) {
   for (int i = 0; i < n; i++) {
       for (int j = 0; j < n / 2; j++) {
           char tmp = arr[i][j];
           arr[i][j] = arr[i][n - j-1];
           arr[i][n - j-1] = tmp;
       }
   }
}
bool check(std::vector<std::vector<char>>& arr1, std::vector<std::vector<char>>& arr2)
    {
   bool flag = true;
   for (int i = 0 ; i < n; ++i) {
       for (int j = 0; j<n; ++j) {
           if (arr1[i][j]!=arr2[i][j]) {
               return false;
```

```
            }
        }
    }
    return true;
}
void print(std::vector<std::vector<char>>& arr1) {
// for (int i = 0 ; i < n; ++i) {
// for (int j = 0; j<n; ++j) {
// std::cout << arr1[i][j];
// }
// std::cout << std::endl;
// }
// std:: cout << std::endl;
}
int main() {

    std::ios_base::sync_with_stdio(false);
    std::cin.tie(0);
    std::cin >> n;
    std::unordered_map<std::string, int> files;
    std::vector<std::string> order;
    std::string s;
           std::getline(std::cin, s);

    for (int i = 0; i < n; ++i) {

        std::getline(std::cin, s);
        std::string exten;
        exten = s.substr(s.find_last_of(".") + 1);
        if (files.find(exten)==files.end()) {
            order.push_back(exten);
            files[exten] = 1;
        } else {
            files[exten]++;
        }
    }
    for (int i = 0; i < order.size(); ++i) {
        std::cout << order[i] << ": " << files[order[i]] << std::endl;
    }
    return 0;
}
```

Grand Prix of Weihai 08.11.2020

Problem status summary

Short name	Long name	Input file name	Output file name	Time limit	Language TL's	Memory limit	Stack limit	Source limit	Submits left	Status	Failed test	Run ID
B	Bridge	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	1997	OK		690
C	Combine The Gears	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	2000			
E	Evil Problemsetters	**input.txt** or *standard input*	**output.txt** or *standard output*	4000 ms		1024 M	64 M	65536 bytes	2000			
G	Game With Stones	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	2000			
H	Hotels	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		256 M	256 M	65536 bytes	2000			
I	Internet Chats	**input.txt** or *standard input*	**output.txt** or *standard output*	7000 ms		256 M	256 M	65536 bytes	1995	OK		1231
J	Julius Caesar and Kazusa	**input.txt** or *standard input*	**output.txt** or *standard output*	20000 ms		256 M	256 M	65536 bytes	2000			
L	Light Version Of Famous Task	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		256 M	256 M	65536 bytes	1994	Time-limit exceeded	16	1594
M	Maximize Minimal Pair Rating	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	1999	OK		850
N	Negative People	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	1997	OK		911
O	Order Backward	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	1999	OK		799
P	Play The Guessing Game	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		256 M	256 M	65536 bytes	1999	OK		743

This is **ejudge** contest administration system, version 2.3.27 #245, compiled Sun Feb 23 04:22:43 2020.

Решение B

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    size_t tests;
    std::cin >> tests;
    std::vector<size_t> ans;
    for (auto i = 0; i < tests ; ++i){
        size_t x,t,n;
        std::cin >> n >> x >> t;
        size_t nowT = 0;
        nowT +=2*t*n;
        size_t nowT1 = nowT;
        nowT1 = std::max(x+2*t,nowT);
        nowT=std::max(x+t,nowT+t);
        nowT= std::min(nowT,nowT1);


        nowT+=t*2*n;
        ans.push_back(nowT);

    }
    for (auto i = 0; i<tests; ++i ) {
```

```
        std::cout << ans[i] << std::endl;
    }
    return 0;
}
```

Решение I

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    std::vector<std::vector<int>> groups;
    int n, m, s;
    std::vector<std::unordered_map<int,int>> groups_ind;
    std::vector<int> groups_messages;
    std::cin >> n >> m >> s;
    groups.resize(n,std::vector<int>());
    groups_messages.resize(n,0);
    groups_ind.resize(n,std::unordered_map<int,int>());
    int t, x, y;
    std::vector<int> students(m,0);

    for (int i = 0; i<s; ++i) {

        std::cin >> t >> x >> y;
        if (t==1) {
            x--;
            y--;
            students[x]-=groups_messages[y];
            groups_ind[y][x] = groups[y].size();
            groups[y].push_back(x);
        }
        if (t==2) {
            y--;
            x--;
            students[x]+=groups_messages[y];

```

```
            int c = groups[y][groups[y].size()-1];
            groups[y][groups[y].size()-1] = groups[y][groups_ind[y][x]];
            groups[y][groups_ind[y][x]] = c;
            groups[y].pop_back();
            groups_ind[y][c] = groups_ind[y][x];
            groups_ind[y].erase(x);
        }
        if (t==3) {
            x--;
            y--;
            students[x]--;
            groups_messages[y]++;
        }
    }
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < groups[i].size(); ++j) {
            students[groups[i][j]]+=groups_messages[i];
        }
    }
    for (int i = 0 ; i < m; ++i) {
        std::cout << students[i] << std::endl;
    }
    return 0;
}
```

Решение L

```
#include <iostream>

#include <vector>

#include <bits/stdc++.h>

#define ull unsigned long long
int main() {

  using namespace std;

  ull T, n;
  cin >> T;
  for (ull i = 0; i < T; ++i) {
    cin >> n;

    std::set <ull> primes;
    bool flag = false;
    while (n % 2 == 0) {

      if (primes.find(2)!=primes.end()) {
        flag = true;
          }
```

```
          primes.insert(2);

      n = n / 2;
    }

    for (int i = 3; i <= sqrt(n) && !flag; i = i + 2) {
      while (n % i == 0) {
          if (primes.find(i)!=primes.end()) {
            flag = true;
          }
         primes.insert(i);
        n = n / i;
      }
    }

    if (n > 2) {
       if (primes.find(n)!=primes.end()) {
              flag = true;
         }
       primes.insert(n);

    }
    if (flag) {
        std::cout << "yes"<< std::endl;
    } else {
        std::cout << "no" << std::endl;
    }

  }
}
```

Решение M

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
```

```
    std::vector<ll> parts;
    ll n;
    std::cin >> n;
    ll a;
    for (auto i = 0; i < n; ++i) {

        std::cin >> a;
        parts.push_back(a);
    }
    std::sort(parts.begin(),parts.end());
    ll min = parts[0]+parts[parts.size()-1];
    for (int i = 0; i < parts.size()/2;++i) {
        if (i!=parts.size()-1-i) {
            ll sum = parts[i]+parts[parts.size()-i-1];
            if (sum<min) {
                min = sum;
            }
        }
    }
    std::cout << min;
    return 0;
}
```

Решение N

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    ll t;
    std::cin >> t;
    std::vector<ll> ans;
    for (auto i = 0; i < t; ++i ) {
        ll m;
        ll min = 0;
        ll p1,p2;
        ll cur = 0;
```

```
        std::cin >> m;
        for (auto j = 0; j<m; ++j) {
            std::cin >> p1 >> p2;
            cur+=p1;
            cur-=p2;
            if (cur<min) {
                min = cur;
            }
        }
        ans.push_back(abs(min));
    }
    for (auto i = 0; i < t; ++i) {
        std::cout << ans[i] << std::endl;
    }
    return 0;
}
```

Решение O

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    ll n;
    std::cin >> n;
    std::vector<std::string> strings;
    std::string str;
    std::getline(std::cin, str);
    for (auto i = 0; i < n; ++i) {
        std::getline(std::cin,str);
        for (auto j = 0; j<str.size()/2; ++j) {
            char c = str[j];
            str[j] = str[str.size()-1-j];
            str[str.size()-1-j] = c;
        }
        strings.push_back(str);
    }
```

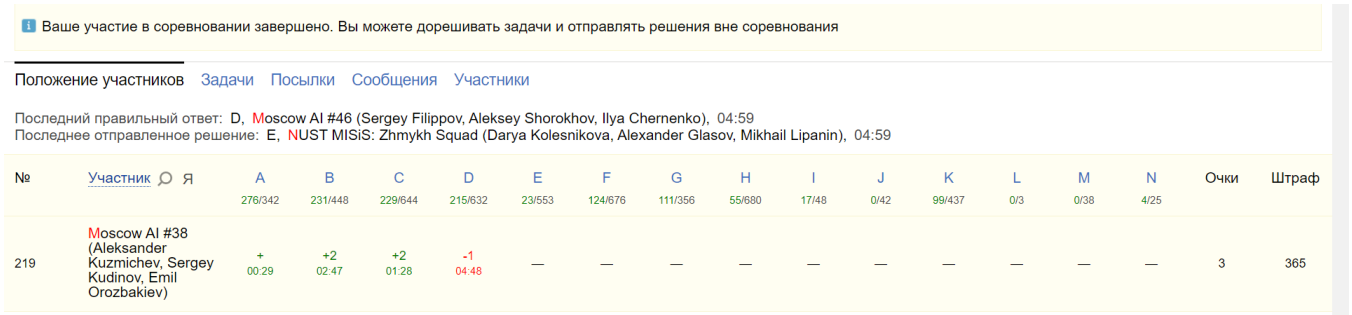
```
    std::sort(strings.begin(),strings.end());
    for (auto i = 0; i < n; ++i) {
        std::cout << strings[i] << std::endl;
    }
    return 0;
}
```

Решение P

```
#include <bits/stdc++.h>
#define ll long long


// ll robot_time(ll x1,ll y1, ll n) {
// if (n<=0) {
// return 0;
// }
// return (x1+y1)*(n-1)+x1;
// }

// ll all_time(ll x1, ll y1, ll n1,ll x2, ll y2, ll n2) {
// return robot_time(x1,y1,n1)+robot_time(x2,y2,n2);
// }
int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    ll n,x,y;
    std::cin >> n >> x >> y;
    ll l,r;
    l = x;
    r = y;
    ll count = 0;
    ll m =0;
    while (r-l>1) {
        m = l+(r-l)/2;
        count++;
        if (m==n) {
            std::cout << count;
            return 0;
        }
        if (m<n) {
            l = m;
        } else
        {
            r = m;
        }

    }
    count++;
    std::cout << count;
```

```
    return 0;
}
```

MRC-2020. Main Contest 15.11.2020



Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

Положение участников Задачи Посылки Сообщения Участники

Последний правильный ответ: D, Moscow AI #46 (Sergey Filippov, Aleksey Shorokhov, Ilya Chernenko), 04:59
Последнее отправленное решение: E, NUST MISiS: Zhmykh Squad (Darya Kolesnikova, Alexander Glasov, Mikhail Lipanin), 04:59

№	Участник	A 276/342	B 231/448	C 229/644	D 215/632	E 23/553	F 124/676	G 111/356	H 55/680	I 17/48	J 0/42	K 99/437	L 0/3	M 0/38	N 4/25	Очки	Штраф
219	Moscow AI #38 (Aleksander Kuzmichev, Sergey Kudinov, Emil Orozbakiev)	+ 00:29	+2 02:47	+2 01:28	-1 04:48	—	—	—	—	—	—	—	—	—	—	3	365

Решение A

```
using namespace std;
#include <bits/stdc++.h>
#define ull unsigned long long


int getSquare(int x, int y)
{
    if (x>0 && y>0) {
        return 1;
    }
    if (x>0 && y<0) {
        return 4;
    }
    if (x<0 && y>0) {
        return 2;
    }
    if (x<0 && y<0) {
        return 3;
    }
    return 0;
}


int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    int x1, y1, x2, y2;
    std::cin >> x1 >> y1 >> x2 >> y2;
    int sq1= getSquare(x1,y1);
    int sq2 = getSquare(x2,y2);
    if (sq1==sq2) {
        std::cout << 0;
```

```
        return 0;
    } else {
        std::cout << 1;
    }
    return 0;
}
```

Решение B

```
players = []
locked = []
SCORES = []

for i in range(2):
  team_ = input()
  for i in range(5):
    players.append((input(), team_))
    #(name, team)
    SCORES.append(0)

n = int(input())

for i in range(n):
  event = input().split()
  team = event[1]
  if (event[2] == "scored"):
    score = int(event[3])
    for i in range(len(players)):
      if (players[i][1] == team):
        SCORES[i] += score
      else:
        SCORES[i] -= score
    for i in range(len(locked)):
      if (locked[i][1] == team):
        SCORES[locked[i][0]] -= score
      else:
        SCORES[locked[i][0]] += score

  if (event[2] == "replaced"):
    Pout = event[3]
    Pin = event[5]
    flag = False
    for i in range(len(locked)):
      if (players[locked[i][0]][0] == Pin and locked[i][1] == team):
        locked.pop(i)
        flag = True
        break
    if (flag == False):
      players.append((Pin, team))
      SCORES.append(0)
```

```
42
43      for i in range(len(players)):
44        if (players[i][0] == Pout and players[i][1] == team):
45          locked.append((i, team))
46          #(index, team)
47
48  for i in range(len(players)):
49    print(players[i][0], end = ' ')
50    print('(' + players[i][1] + ')', end = ' ')
51    if (SCORES[i] > 0):
52      print('+' + str(SCORES[i]))
53    elif (SCORES[i] < 0):
54      print(str(SCORES[i]))
55    else:
56      print(0)
```

Решение C

```
1  import java.lang.Long.parseLong
2
3  fun sum(a:Int, b:Int) : Int {
4      return a+b
5  }
6  private fun readLn() = readLine()!!
7  private fun readInt() = readLn().toInt()
8  private fun readStrings() = readLn().split(" ")
9  private fun readInts() = readStrings().map { it.toInt() }
10 fun main(){
11     var a = readLine()!!
12     a = a.toUpperCase()
13     var strs = a.split(" ")
14     var isPascal = true
15     var isC = true
16     var sizeStr = -1L
17     var size1 = 0L
18     var l = 2*16L
19
20     var isZeroByte = false
21     var r = parseLong("7F",16)
22     for (str in strs) {
23         var sum1 = parseLong(str,16)
24         if (sizeStr==-1L) {
25             sizeStr = sum1
26             if (sizeStr>255) {
27                 isPascal = false
28             }
29             if (sum1==0L) {
30                 isZeroByte = true
31             }
32             if (!isZeroByte && (sum1 !in l..r)) {
```

```
                isC = false
            }
        } else {
            size1++
            if ((sum1 !in l..r) && size1<=sizeStr) {
               isPascal = false
            }
            if (sum1==0L) {
                isZeroByte = true
            }
            if (!isZeroByte && (sum1 !in l..r)) {
                isC = false
            }
        }

    }
    if (isC) {
        isC = isZeroByte
    }
    if (isPascal) {
        if (size1<sizeStr) {
            isPascal = false
        }
    }
    if (isC && isPascal) {
        println("Both")
    }
    if (!isC && !isPascal) {
        println("None")
    }
    if (isC && !isPascal) {
        println("C")
    }
    if (!isC && isPascal) {
        println("Pascal")
    }
}
```

Решение D

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;


int main()
{
    int a[9] = {2, 3, 5, 7, 23, 37, 53, 73, 373};
    int answer = 0;
```

```
    long long n, m;
    cin >> m >> n;
    for (int i = 0; i < 9; i ++){
        if ((a[i] >= m) && (a[i] <= n)){
            answer++;
        }
    }
    cout << answer << endl;
}
```

Regional 1. Northwestern Russia 2020 22.11.2020

48	MAI #38 Kudinov, Kuzmichev, Orozbakiev	+ 00:05	+ 00:20	—	—	—	—	—	—	+2 01:02	—	—	—	—	—	3	128

Решение A

```
#include <bits/stdc++.h>


using namespace std;

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    int n;
    std::cin >> n;
    if (n==1996 || n==1997 || n==2000 || n==2007 || n==2008 || n == 2013 || n==2018) {
        std::cout << "SPbSU";
        return 0;
    }
    if (n==2006) {
        std::cout <<"PetrSU, ITMO";
        return 0;
    }
    std::cout << "ITMO";
    return 0;
}
```

Решение B

```
#include <bits/stdc++.h>

#define ull unsigned long long
using namespace std;

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    long long a,b,x,y, t;
```

```
    std::cin >> a >> x >> b >>y >> t;
    long long res1,res2;
    res1=0;
    res2=0;
    long long t1 = t-30;
    long long t2 = t-45;
    if (t1>0) {
        res1 = res1+t1*x;

    }
     if (t2>0) {
        res2 = res2+t2*y;

    }
    res1*=21;
    res2*=21;
    res1+=a;
    res2+=b;
    std::cout << res1 << " " << res2;
    return 0;
}
```

Решение I

```
#include <bits/stdc++.h>

#define ull unsigned long long
using namespace std;

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
   long long s;
   std::cin >> s;
   for (auto i = 0; i<s; ++i) {
       long long a = sqrt(i);
       long long b = sqrt(s-i);
       if (a*a+b*b==s) {
           if (a!=0) {
           std::cout << 0 << " " << b << std::endl;
           std::cout << a << " " << 0 << std::endl;
           std::cout << a+b << " " << a << std::endl;
           std::cout << b << " " << a+b << std::endl;
            return 0;
           }
           else {
            std::cout << b << " " << 0 << std::endl;
           std::cout << 0 << " " << 0 << std::endl;
           std::cout << 0 << " " << b << std::endl;
           std::cout << b << " " << b << std::endl;
```

```
            return 0;
            }
        }
    }
    std::cout << "Impossible";
     return 0;
}
```

Grand Prix of NorthBeach 29.11.2020

Problem status summary

Short name	Long name	Input file name	Output file name	Time limit	Language TL's	Memory limit	Stack limit	Source limit	Submits left	Status	Failed test	Run ID
A	Arrange And Count	**input.txt** or *standard input*	**output.txt** or *standard output*	7000 ms		512 M	512 M	65536 bytes	2000			
B	Build More 2020's	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		512 M	512 M	65536 bytes	2000			
C	Choose Two Subsequences	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		512 M	512 M	65536 bytes	2000			
D	Determinant Strikes Back	**input.txt** or *standard input*	**output.txt** or *standard output*	5000 ms		512 M	512 M	65536 bytes	2000			
E	Efficient Data Structure	**input.txt** or *standard input*	**output.txt** or *standard output*	4000 ms		512 M	512 M	65536 bytes	2000			
H	Hamming Distance	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		512 M	512 M	65536 bytes	2000			
I	Integers and Ranges	**input.txt** or *standard input*	**output.txt** or *standard output*	7000 ms		512 M	512 M	65536 bytes	2000			
K	Autonomous Cities	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		512 M	512 M	65536 bytes	1999	OK		237
L	Customity	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		512 M	512 M	65536 bytes	1994	OK		467
M	Interesting Points	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		512 M	512 M	65536 bytes	1999	OK		262

Решение K

```
#include <bits/stdc++.h>

#define ull unsigned long long
using namespace std;

bool fun(size_t n, size_t k) {
    // std::cout << k << std::endl;
    size_t t = 0;
    if (k<1) {
        t = 0;
    } else {
        t = k*(k-1)/2;
        // std::cout << t;
        // std::cout << (n-1-n/2) << std::endl;
    }
    if (t<(n-1-n/2)) {
        return true;
        // std::cout << k;
    }
    return false;
}

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    size_t n;
    std::cin >> n;
```

```
    n-=n%2;
    size_t l,r,m;
    l = 0;
    r = n;
    size_t ans;
    while (r-l>1) {
        m = l+(r-l)/2;
        if (fun(n,m)) {
            l = m;
        } else {
            r = m;
        }
    }
   if (!fun(n,r)) {
        ans = r;
    }
    if (!fun(n,m)) {
        ans = m;
    }

      if (!fun(n,l)) {
        ans = l;
    }
    // if (!fun(n,1)) {
    // ans = 1;
    // }

    std::cout << 2;
    return 0;
}
```

Решение L

```
#include <iostream>
#include <bits/stdc++.h>
#include <cmath>


using namespace std;


int main() {
    long long N;
    cin >> N;
    long long a[N];
    long long max;

    cin >> a[0];
    max = a[0];
    for (long long i = 1; i < N; ++i) {
```

```
        cin >> a[i];
        if(a[i] > max){
            max = a[i];
        }
    }

    long long mind = abs(a[1] - a[0]);



    long long count = 0;

        for (long long i = 0; i < N; ++i) {
            if (a[i] == max){
                count++;
            }
    }

        cout << N - count << endl;

}
```

Решение M

```
#include <bits/stdc++.h>

#define ull unsigned long long
using namespace std;

bool fun(size_t n, size_t k) {
    // std::cout << k << std::endl;
    size_t t = 0;
    if (k<1) {
        t = 0;
    } else {
        t = k*(k-1)/2;
        // std::cout << t;
        // std::cout << (n-1-n/2) << std::endl;
    }
    if (t<(n-1-n/2)) {
        return true;
        // std::cout << k;
    }
    return false;
}

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(NULL);
    size_t n;
```

```
27      std::cin >> n;
28      std::cout << n+1;
29      return 0;
30  }
```