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
Орозбакиев Эмиль Данияpович	М8О-306Б-18
Команда:	MAI #38: Kuzmichev, Orozbakiev

Москва, 2021

Сводная таблица за весну 2021

Дата	Название	Время	Место проведения	Решенные задачи
14.02.2021	Grand Prix of Belarus	11:00	Дистанционно	N, P
21.02.2021	Grand Prix of Suwon	11:00	Дистанционно	M
28.02.2021	Grand Prix of Tokyo	11:00	Дистанционно	N
14.03.2021	20-21 своя тренировка 2: Semi-FFT	12:00	Дистанционно	F, G, H, I, J
21.03.2021	20-21 своя тренировка 3: Graphs	11:00	Дистанционно	C, F, I, J, K
11.04.2021	20-21 своя тренировка 4	11:00	Дистанционно	B, D, F, L, M
25.04.2021	RuCode 3.0	11:00	Дистанционно	A
09.05.2021	Grand Prix of Urals	11:00	Дистанционно	A, N

Явка на контесты

Дата	Название	Присутствующие
14.02.2021	Grand Prix of Belarus	Кузьмичев, Орозбакиев
21.02.2021	Grand Prix of Suwon	Кузьмичев, Орозбакиев
28.02.2021	Grand Prix of Tokyo	Кузьмичев, Орозбакиев
14.03.2021	20-21 своя тренировка 2: Semi-FFT	Кузьмичев, Орозбакиев
21.03.2021	20-21 своя тренировка 3: Graphs	Кузьмичев, Орозбакиев
11.04.2021	20-21 своя тренировка 4	Кузьмичев, Орозбакиев
25.04.2021	RuCode 3.0	Кузьмичев, Орозбакиев
09.05.2021	Grand Prix of Urals	Кузьмичев, Орозбакиев

Решения задач

Grand Prix of Belarus 14.02.2021

Short name	Long name	Input file name	Output file name	Time limit	Language
B	Best Meeting Places	**input.txt** or *standard input*	**output.txt** or *standard output*	5000 ms	
C	Colorful Squares	**input.txt** or *standard input*	**output.txt** or *standard output*	8000 ms	
E	Expected Distance	**input.txt** or *standard input*	**output.txt** or *standard output*	6000 ms	
F	Find the XOR	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms	
G	Generate The Array	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms	
	Integer Array Shuffle	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms	
J	Junkyeom's Contest	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms	
M	Move To The Equality	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms	
N	Numbers	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms	
O	Olympic Tournament	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms	
P	Product Game	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms	
Q	Quantum Robot	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms	

Решение 1

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    long long a, b, c, d, a1, b1, c1, d1;
    while(true){
        cin >> a >> b >> c >>d;
        int k = 0;
        if (a == 0){
            break;
        } else {
            while (true){
                if(a == b){
                    if(b == c){
                        if (c == d){
                            if (d == a){
                                break;
                            }
                        }
                    }
                }
                k += 1;
                a1 = abs(a - b);
                b1 = abs(b - c);
                c1 = abs(c - d);
                d1 = abs(d - a);
```

```
                a = a1;
                b = b1;
                c = c1;
                d = d1;

            }
            cout << k << endl;
        }
    }
}
```

Решение 2

```
#include <iostream>
#include <ctime>

using namespace std;

long long digitsSum(long long n) {
    long long sum = 0;
  while (n != 0) {
    sum += n % 10;
    n /= 10;
  }
  return sum;
}


long long digitsProduct(long long n) {
    long long sum = 1;
    while (n != 0) {
        sum *= n % 10;
        n /= 10;
    }
    return sum;
}

long long num_of_digits(long long n){
    long long sum = 1;
    while (n > 9){
        sum *= 10;
        n /= 10;
    }
    sum -= 1;
    return sum;
}


int main() {

```

```
    long long n, k, z;


    long long max_sum = 0, max = 0, answer = 0;

    cin >> n;



    if (n > 9 && n < 1000000000){

    k = num_of_digits(n);

    for (long long i = k; i <=n; ++i) {
        z = digitsProduct(i);
        if(z > max_sum){
            max_sum = z;
            max = i;
        }
    }


        std::cout << max_sum << std::endl ;


    } else if(n >= 1000000000){
        cout << 387420489 << endl;
    } else if(n < 10) {
        cout << n;
    }
    return 0;
}
```

Решение 7 (Не засчитано)

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;

struct skater {
    int x;
    int y;
    int number;
};

bool comp(const skater a, const skater b) {
    return (a.x < b.x) || (a.x == b.x && a.y < b.y);
```

```
};

int main() {
    int n;
    skater a;
    vector<skater> skaters;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> a.x >> a.y;
        a.number = i + 1;
        skaters.push_back(a);
    }
    for (int i = 0; n / 2 > i; ++i) {
        int min_dist = -1;
        int number;
        int del;
        for (int j = 1; j < skaters.size(); j++) {
            int x, y;
            if (skaters.front().x > skaters[j].x) {
                x = skaters.front().x - skaters[j].x;
            } else {
                x = skaters[j].x - skaters.front().x;
            }
            if (skaters.front().y > skaters[j].y) {
                y = skaters.front().y - skaters[j].y;
            } else {
                y = skaters[j].y - skaters.front().y;
            }
            int dist = pow(x, 2) + pow(y, 2);
            if (min_dist < 0) {
                min_dist = dist;
                number = skaters[j].number;
                del = j;
            } else if (min_dist > dist) {
                min_dist = dist;
                number = skaters[j].number;
                del = j;
            }
        }
        cout << skaters.front().number << ' ' << number << endl;
        skaters.erase(skaters.begin() + del);
        skaters.erase(skaters.begin());
    }
    return 0;
}
```

Grand Prix of Suwon 21.02.2021

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	Best Meeting Places	**input.txt** or *standard input*	**output.txt** or *standard output*	5000 ms		1024 M	64 M	65536 bytes	2000			
C	Colorful Squares	**input.txt** or *standard input*	**output.txt** or *standard output*	8000 ms		1024 M	64 M	65536 bytes	2000			
E	Expected Distance	**input.txt** or *standard input*	**output.txt** or *standard output*	6000 ms		1024 M	64 M	65536 bytes	2000			
F	Find the XOR	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
G	Generate The Array	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
I	Integer Array Shuffle	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	2000			
J	Junkyeom's Contest	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
M	Move To The Equality	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	1998	OK		257
N	Numbers	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
O	Olympic Tournament	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	1999	Wrong answer	1	463
P	Product Game	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	1997	Time-limit exceeded	8	316
Q	Quantum Robot	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			

Решение M

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    long long a, b, c, d, a1, b1, c1, d1;
    while(true){
        cin >> a >> b >> c >>d;
        int k = 0;
        if (a == 0){
            break;
        } else {
            while (true){
                if(a == b){
                    if(b == c){
                        if (c == d){
                            if (d == a){
                                break;
                            }
                        }
                    }
                }
                k += 1;
                a1 = abs(a - b);
                b1 = abs(b - c);
                c1 = abs(c - d);
                d1 = abs(d - a);
                a = a1;
                b = b1;
                c = c1;
                d = d1;

            }
            cout << k << endl;
        }
    }
}
```

Решение O (не засчитано)

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;

struct skater {
    int x;
    int y;
    int number;
};

bool comp(const skater a, const skater b) {
    return (a.x < b.x) || (a.x == b.x && a.y < b.y);
};

int main() {
    int n;
    skater a;
    vector<skater> skaters;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> a.x >> a.y;
        a.number = i + 1;
        skaters.push_back(a);
    }
    for (int i = 0; n / 2 > i; ++i) {
        int min_dist = -1;
        int number;
        int del;
        for (int j = 1; j < skaters.size(); j++) {
            int x, y;
            if (skaters.front().x > skaters[j].x) {
                x = skaters.front().x - skaters[j].x;
            } else {
                x = skaters[j].x - skaters.front().x;
            }
            if (skaters.front().y > skaters[j].y) {
                y = skaters.front().y - skaters[j].y;
            } else {
                y = skaters[j].y - skaters.front().y;
            }
            int dist = pow(x, 2) + pow(y, 2);
            if (min_dist < 0) {
                min_dist = dist;
                number = skaters[j].number;
                del = j;
```

```
            } else if (min_dist > dist) {
                min_dist = dist;
                number = skaters[j].number;
                del = j;
            }
        }
        cout << skaters.front().number << ' ' << number << endl;
        skaters.erase(skaters.begin() + del);
        skaters.erase(skaters.begin());
    }
    return 0;
}
```

Решение P (не засчитано)

```
#include <iostream>

using namespace std;

int digitsSum(int n) {
  int sum = 0;
  while (n != 0) {
    sum += n % 10;
    n /= 10;
  }
  return sum;
}


int digitsProduct(int n) {
    int sum = 1;
    while (n != 0) {
        sum *= n % 10;
        n /= 10;
    }
    return sum;
}

int num_of_digits(int n){
    int sum = 9;
    while (n > 100){
        sum *= 10;
        n /= 10;
    }
    return sum;
}


int main() {

```

```
    int n, k;


    int max_sum = 0, max = 0, answer = 0;

    cin >> n;

    if (n > 9){

    k = num_of_digits(n);

    for (int i = k; i <= n; ++i) {
        if(digitsProduct(i) > max_sum){
            max_sum = digitsProduct(i);
            max = i;
        }
    }

    std::cout << max_sum << std::endl;
        } else {
        cout << n;
    }
    return 0;
}
```

Grand Prix of Tokyo 28.02.2021

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	Bit Operation	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		1024 M	64 M	65536 bytes	2000			
E	Edge Subsets	**input.txt** or *standard input*	**output.txt** or *standard output*	9000 ms		1024 M	64 M	65536 bytes	2000			
G	Games	**input.txt** or *standard input*	**output.txt** or *standard output*	6000 ms		1024 M	64 M	65536 bytes	2000			
H	Harsh Comments	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
I	Inverse Problem	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		1024 M	64 M	65536 bytes	2000			
K	King and Highways	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	2000			
L	Looking For Plagiarism	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	1997	Wrong answer	3	542
M	Megafactorial	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	1997	Time-limit exceeded	11	287
N	N-Dimensional Game	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	1994	OK		289
O	Odds For Palindrome	**input.txt** or *standard input*	**output.txt** or *standard output*	1000 ms		1024 M	64 M	65536 bytes	2000			

Решение N

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;

long long stepen(long long base, long long p) {
    if (p == 1) {
        return base;
    }
```

```
12
13      if (p % 2 == 0) {
14          long long t = stepen(base, p / 2);
15          return t * t % 998244353;
16      } else {
17          return stepen(base, p - 1) * base % 998244353;
18      }
19  }
20
21  int main() {
22      long long n;
23      long long a = 0, b = 0, c = 0, d = 0;
24      cin >> n;
25      if(n == 3){
26          cout << 60;
27          return 0;
28      } else if (n == 4){
29          cout << 5 + 8 + 16 + 24 + 120 + 600;
30      } else {
31          a = (n + 1) % 998244353;
32          b = (n * 2) % 998244353;
33          d = (a + b) % 998244353;
34          c = stepen(2, n);
35          cout << (d + c) % 998244353;
36      }
37  }
```

Решение M (не засчитано)

```
1   #include <iostream>
2   #include <vector>
3   #include <algorithm>
4   #include <cmath>
5
6   using namespace std;
7
8   long long factoria(long long n){
9       long long fact = 1;
10      while (n > 0){
11          fact *= n;
12          fact %= 998244353;
13          n--;
14      }
15      return fact;
16  };
17
18  int main() {
19      long long n;
20      cin >> n;
21      long long ans;
```

```
    ans = factoria(n);
    long long otvet;
    otvet = factoria(ans);
    cout << otvet;
}
```

Решение L (не засчитано)

```
#include <iostream>

using namespace std;

long long stepen(long long base, long long p) {
    if (p == 1) {
        return base;
    }

    if (p % 2 == 0) {
        long long t = stepen(base, p / 2);
        return t * t % 998244353;
    } else {
        return stepen(base, p - 1) * base % 998244353;
    }
}

int main() {
    long long r = 1;
    long long n;
    cin >> n;
    if (n == 2){
        cout << 1;
        return 0;
    } else if(n == 4){
        cout << 2;
    } else{ n /=2;

    r = stepen(2, n);
    if(n % 2 == 1){
        r = (r - 2) % 998244353;
    } else {
        r = (r + 2) % 998244353;
    }
    cout << r;
    }

}
```

20-21 своя тренировка 2: Semi-FFT 14.03.2021

19	MAI #35 Korostelev, Trofimov, Cheremisinov	4	580				-3		+3 01:44	+ 00:12	+6 03:29	-12	+ 01:15
20	MAI #47 Arkhipov, Poley-Dobronravova, Essaulov	4	658						+4 02:33	+ 00:21		+2 03:08	+3 01:56
21	MAI #22 Ilarionov, Firfarov, Sharapov	4	664						+2 02:49	+ 01:12	+2 02:58	-1	+4 01:25
22	MAI #24 Bakharev, Sindyukov	4	702						+3 02:47	+ 00:36		+1 00:44	+7 03:55
23	MAI #38 Kuzmichev, Orozbakiev	3	264							+ 00:32	+ 02:18	-2	+ 01:34
24	MAI #28 Batyanovsky, Veprintsev, Tyuneev	3	308						+ 02:13	+ 01:33			+ 01:22
25	MAI #44 Vakhramyan, Kosogorov, Simonov	3	377							+ 00:36		+4 01:02	+3 02:19
26	MAI #40 Dubovitsky, Kuzmin, Leshkova	3	449						-5	+ 00:39	+2 01:38	-1	+3 03:32
27	MAI #25 Vasiliev, Ilyin, Marochkin	3	500						+2 03:50	+1 00:25			+1 02:45
28	MAI #30 Dergach, Moskalenko, Khasanov	2	387							+ 01:58	-6		+3 03:29
29	MAI #26 Krassotkin, Revunov, Fedorov	1	50		-1					+ 00:50	-4		-1
30	MAI #36 Vankov, Dubinin, Poskryakov	1	181						-1	+2 02:21			
	* MAI #38 Kuzmichev, Orozbakiev	2							+			+1	
	* MAI #35 Korostelev, Trofimov, Cheremisinov	1										+	
	* MAI #23 Kudinov, Leychenko, Markov	0							-2				
	* MAI #24 Bakharev, Sindyukov	0											

Решение F

```cpp
#include<cstdio>
#include<iostream>
#include<cstring>
using namespace std;
const int maxn=1e5+5;
char a[maxn],b[maxn],aa[maxn],bb[maxn];
int main(){
  int T;
  scanf("%d",&T);
  while(T--){
    int n;
    scanf("%d",&n);
    scanf("%s",a+1);
    scanf("%s",b+1);
    int t=0;
    for(int i=1;i<=n;i++){
      if(a[i]!=b[i]){
        aa[t]=a[i];
        bb[t]=b[i];
        t++;
      }
    }
    bool flag=true;
    for(int i=0;i<t;i++){
      if(bb[i]==aa[(i+1)%t])
      continue;
      else{
```

```
        flag=false;
        break;
      }
    }
    if(flag) printf("YES\n");
    else printf("NO\n");
  }
  return 0;
}
```

Решение G

```
#include <iostream>

using namespace std;


int main() {
    int N;
    long long n, x, y;
    cin >> N;
    for (int i = 0; i < N; ++i) {
        cin >> n >> x >> y;
        if(n % 2 == 0){
            if(n/2 <= x){
                if(n/2 <= y){
                    cout << "YES" << endl;
                } else {cout << "NO" << endl;}
            }else {cout << "NO" << endl;}
        } else {
            if(n/2+1<=x){
                if(n/2 <= y){
                    cout << "YES" << endl;
                }else {cout << "NO" << endl;}
            }else {cout << "NO" << endl;}
        }
    }
}
```

Решение H

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    int T;
    cin >> T;
    for (int i = 0; i < T; ++i) {
        long long a, b;
```

```
        vector<long long> rep_a;
        vector<long long> rep_b;
        cin >> a >> b ;
        int pow = 1;
        while(a != 0){
            rep_a.push_back((a % 10) * pow);
            a /= 10;
            pow *= 10;
        }

        pow = 1;

        while(b != 0){
            rep_b.push_back((b % 10) * pow);
            b /= 10;
            pow *= 10;
        }
        for (int j = 0; j < rep_a.size(); ++j) {
            for (int k = 0; k < rep_b.size(); ++k) {
                if(rep_a[j] != 0 && rep_b[k] != 0){
                    if(j == rep_a.size() - 1 && k == rep_b.size() - 1){
                        cout << rep_a[j] << " " << "x" << " " << rep_b[k] << endl;
                    }else {
                        cout << rep_a[j] << " " << "x" << " " << rep_b[k] << " " << "+"
                             << " ";
                    }
                }
            }
        }
    }
    return 0;
}
```

Решение I

```
#include <iostream>
#include <cstring>


using namespace std;


int main() {
    long long T;
    cin >> T;
    for (long long i = 0; i < T; ++i) {
        long long n, m, k, ans = 0;
        long long alphabet[26][26];
        long long counter[26];
        memset(alphabet, 0, sizeof(alphabet));
```

```
        memset(counter, 0, sizeof(counter));
        cin >> n >> m >> k;
        string a, b;
        cin >> a;
        cin >> b;
        for (long long j = 0; j < n; ++j) {
            if(j + k -1 < n){
                alphabet[a[j + k - 1] - 'a'][a[j]-'a']= 1;
            }
        }
        for (long long j = 0; j < m; ++j) {
            counter[b[j] - 'a']++;
            for (long long l = 0; l <26; ++l) {
                if(alphabet[b[j]-97][l]){
                    ans+=counter[l];
                }
            }
        }
        cout << ans << endl;
    }
    return 0;
}
```

Решение G

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main() {
    int T;
    cin >> T;

    for (int i = 0; i < T; ++i) {
        long long n, m, k;
        cin >> n >> m >> k;
        char a, b;
        long long counter = 0, c;
        vector<int> siz;
        for (int j = 0; j < n; ++j) {
            cin >> a;
            if ((a == 'S') && (j < k)) {
                siz.push_back(1);
            } else if (a == 'S') {
                siz.push_back(0);
            } else {
                siz.push_back(-1);
            }
```

```
        }
        for (int j = 0; j < m; ++j) {
            cin >> b;
            if (b == 'W') {
                cin >> c;
                if (siz[c - 1] == 1) {
                    counter++;
                }
                if (siz[c - 1] == 0 || siz[c - 1] == -1) {
                    for (int l = 0; l < c; ++l) {
                        if (siz[l] == 0) {
                            siz[l] = 1;
                        }
                    }
                }
            } else {
                for (int l = 0; l < n; ++l) {
                    if (siz[l] == 0) {
                        siz[l] = 1;
                    }
                }
            }
        }
        cout << counter << endl;
    }
    return 0;
}
```

20-21 своя тренировка 3: Graphs 21.03.2021

25	MAI #48 Burkevich, Kochuikova, Shukhova	0	0									-3	
25	MAI #30 Dergach, Moskalenko, Khasanov	0	0			-3						-3	-5
	* MAI #38 Kuzmichev, Orozbakiev	5				+			+			+	+	+
	* MAI #17 Belousov, Korotkevich, Sazhenov	3						+1			+2	+5		
	* MAI #41 Bronnikov, Veltman, Zhivykh	2							+3					+

Решение C

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

//adjacency list
std::vector<std::list<int>> adjacency_list;
std::vector<bool> visited;
std::vector<bool> leaf;
int res = 0;

void dfs_visit(int u) {
    visited[u] = true;
    int cnt = 0;
    int son = 0;
```

```
    for (int v : adjacency_list[u]) {
        if (!visited[v]) {
            cnt++;
            son = v;
            dfs_visit(v);
        }
    }

    if (cnt == 0) {
        leaf[u] = true;
    } else if (cnt == 1) {
        if (leaf[son]) {
            res++;
        }
    }

}

void dfs() {
    visited[0] = true;
    for (int u : adjacency_list[0]) {
        if (!visited[u]) {
            dfs_visit(u);
        }
    }
}

int main() {
    int n;
    std::cin >> n;
    adjacency_list.resize(n);
    visited.assign(n, false);
    leaf.assign(n, false);

    int from, to;
    for (int i = 0; i < n - 1; i++) {
        std::cin >> from >> to;
        adjacency_list[from - 1].push_back(to - 1);
        adjacency_list[to - 1].push_back(from - 1);
    }

    dfs();
    std::cout << res << "\n";

    return 0;
}
```

Решение F

```
#include <iostream>
```

```
2  #include <ctime>
3  #include <vector>
4
5  using namespace std;
6
7
8  int mark_cc(int i, int j, int n, int m, const std::vector <std::vector <char>>& g, std
       ::vector <std::vector <int>>& used) {
9  if (i < 0 || i >= n) {
10 return 0;
11 }
12
13 if (j < 0 || j >= m) {
14 return 0;
15 }
16
17 if (used[i][j]) {
18 return 0;
19 }
20
21 if (g[i][j] != '#') {
22 return 0;
23 }
24
25 used[i][j] = 1;
26
27 mark_cc(i + 1, j, n, m, g, used);
28 mark_cc(i - 1, j, n, m, g, used);
29 mark_cc(i, j + 1, n, m, g, used);
30 mark_cc(i, j - 1, n, m, g, used);
31
32 return 1;
33 }
34
35 int main()
36 {
37 std::ios::sync_with_stdio(false);
38 int n, m;
39 std::cin >> n >> m;
40 int cnt = 0;
41 std::vector <std::vector <char>> gr(n, std::vector <char> (m));
42 for (int i = 0; i < n; ++i) {
43 std::string s;
44 std::cin >> s;
45 for (int j = 0; j < m; ++j) {
46 gr[i][j] = s[j];
47 if (gr[i][j] == '#') {
48 cnt += 1;
49 }
```

```
}
}

if (cnt < 3) {
std::cout << -1 << '\n';
return 0;
}

for (int i = 0; i < n; ++i) {
for (int j = 0; j < m; ++j) {
if (gr[i][j] == '#') {
std::vector <std::vector <int>> used(n, std::vector <int> (m));
cnt += 1;
gr[i][j] = '.';
int side_steps = 0;
side_steps += mark_cc(i + 1, j, n, m, gr, used);
//std::cout << i << " " << j << " " << side_steps << '\n';
side_steps += mark_cc(i - 1, j, n, m, gr, used);
//std::cout << i << " " << j << " " << side_steps << '\n';
side_steps += mark_cc(i, j + 1, n, m, gr, used);
//std::cout << i << " " << j << " " << side_steps << '\n';
side_steps += mark_cc(i, j - 1, n, m, gr, used);
//std::cout << i << " " << j << " " << side_steps << '\n';

if (side_steps >= 2) {
std::cout << 1 << '\n';
return 0;
}
gr[i][j] = '#';

}
}
}

std::cout << 2 << '\n';
return 0;
}
```

Решение I

```
#include<bits/stdc++.h>
using namespace std;

struct Door {
int locked;
int first_switch = -1;
int second_switch = -1;

int GetIncident(int sw) {
if (sw == first_switch) {
```

```
return second_switch;
}
return first_switch;
}

void AddVertex(int v) {
if (first_switch == -1) {
first_switch = v;
} else {
second_switch = v;
}
}
};

using Graph = vector<vector<Door*>>;


bool bfs(const Graph& g, int start, vector<int>& color) {
queue<int> q;
color[start] = 0;
q.push(start);
while (!q.empty()) {
int u = q.front();
q.pop();
for (Door* door : g[u]) {
int v = door->GetIncident(u);
int door_status = door->locked;
int correct_color = (color[u] + ((door_status + 1) & 1u)) & 1u;
if (color[v] == -1) {
color[v] = correct_color;
q.push(v);
} else if (color[v] != correct_color) {
return false;
}

}
}
return true;
}

int main() {
int n,m;
cin >> n >> m;
vector<Door> doors(n);
for (Door& i : doors) {
cin >> i.locked;
}
Graph g(m);
for (int i = 0; i < m; ++i) {
```

```
int c;
cin >> c;
for (int j = 0; j < c; ++j) {
int door;
cin >> door;
door--;
g[i].push_back(&doors[door]);
doors[door].AddVertex(i);
}
}
vector<int> color(g.size(), -1);

for (int i = 0; i < g.size(); ++i) {
if (color[i] == -1) {
if (!bfs(g, i, color)) {
cout << "NO\n";
return 0;
}
}
}

cout << "YES\n";
}
```

Решение J

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

#define ull unsigned long long

std::vector<std::list<int>> adjacency_list;
std::vector<bool> visited;
std::vector<int> first_set;
std::vector<int> second_set;

void dfs_visit(int u, int set) {
    visited[u] = true;
    if (set == 1) {
        first_set.push_back(u);
    } else {
        second_set.push_back(u);
    }
    for (int v : adjacency_list[u]) {
        if (!visited[v]) {
            if (set == 1) {
                dfs_visit(v, 2);
            } else {
                dfs_visit(v, 1);
            }
        }
```

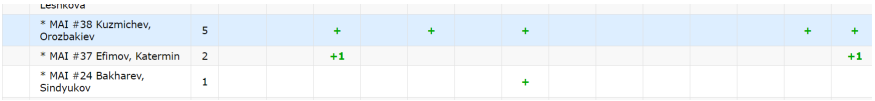
```
    }
}

void dfs() {
    visited[0] = true;
    first_set.push_back(0);
    for (int u : adjacency_list[0]) {
        if (!visited[u]) {
            dfs_visit(u, 2);
        }
    }
}

ull res = 0;

int main() {
    int n;
    std::cin >> n;
    adjacency_list.resize(n);
    visited.assign(n, false);

    int from, to;
    for (int i = 0; i < n - 1; i++) {
        std::cin >> from >> to;
        adjacency_list[from - 1].push_back(to - 1);
        adjacency_list[to - 1].push_back(from - 1);
    }

    dfs();

    if (first_set.size() < second_set.size()) {
        for (int v : first_set) {
            res += second_set.size() - adjacency_list[v].size();
        }
    } else {
        for (int v : second_set) {
            res += first_set.size() - adjacency_list[v].size();
        }
    }

    std::cout << res << "\n";

    return 0;
}
```

Решение K

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

std::vector<std::list<int>> adjacency_list;
```

```cpp
std::vector<bool> visited;
std::vector<int> height;

void dfs_visit(int u, int h) {
    visited[u] = true;
    height[h]++;
    for (int v : adjacency_list[u]) {
        if (!visited[v]) {
            dfs_visit(v, h + 1);
        }
    }
}

void dfs() {
    visited[0] = true;
    height[0] = 1;
    for (int u : adjacency_list[0]) {
        if (!visited[u]) {
            dfs_visit(u, 1);
        }
    }
}

int main() {
    int n;
    std::cin >> n;
    adjacency_list.resize(n);
    visited.assign(n, false);
    height.resize(n);

    int to;
    for (int i = 0; i < n - 1; i++) {
        std::cin >> to;
        adjacency_list[i + 1].push_back(to - 1);
        adjacency_list[to - 1].push_back(i + 1);
    }

    dfs();
    int res = 0;
    for (int i = 0; i < n && height[i] != 0; i++) {
        res += (height[i] % 2);
    }

    std::cout << res << "\n";

    return 0;
}
```

20-21 своя тренировка 4 11.04.2021



	Lesnikova													
	* MAI #38 Kuzmichev, Orozbakiev	5			+		+		+				+	+
	* MAI #37 Efimov, Katermin	2			+1									+1
	* MAI #24 Bakharev, Sindyukov	1							+					

Решение B

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

using namespace std;

int main() {
int n,k;
cin >> n >> k;
if (k == 1 || (k == 2 && n % 2 == 0) || (n == 4 && k == 3)) {
cout << "Yes\n";
} else {
cout << "No\n";
}

return 0;
}
```

Решение D

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

bool in(char ch, const std::vector<char> &vec) {
    for (int i = 0; i < vec.size(); i++) {
        if (ch == vec[i])
            return true;
    }
    return false;
}

int main() {
    long long n;
    std::cin >> n;
    std::string s;
    std::cin >> s;

    std::vector<char> vowel{'a', 'i', 'o', 'u', 'e', 'y'};

    int help = 2;
    long long res = 0;

    for (long long i = 0; i < n; i++) {
        if (help == 2) {
            if (in(s[i], vowel)) {
```

```
                help = 1;
            } else {
                help = 0;
            }
        } else if (help == 0) {
            if (in(s[i], vowel)) {
                help = 2;
                res++;
            }
        } else if (help == 1) {
            if (!in(s[i], vowel)) {
                help = 2;
                res++;
            }
        }
    }

    std::cout << res << "\n";

    return 0;
}
```

Решение F

```
#include <iostream>
#include <vector>

int main() {
    int n;
    std::cin >> n;
    std::vector<int> a(n);

    for (int i = 0; i < n; i++) {
        std::cin >> a[i];
    }

    if (n == 1) {
        std::cout << "Yes\n";
    } else {
        for (int i = 1; i < n - 1; i++) {
            if (a[i] == 2) {
                a[i] = 0;
                if (a[i - 1] - 1 < 0 || a[i + 1] - 1 < 0) {
                    std::cout << "No\n";
                    return 0;
                } else {
                    a[i - 1]--;
                    a[i + 1]--;
                }
            }
```

```
        }
        if (a[0] == 2) {
            a[0] = 0;
            if (a[1] - 1 < 0) {
                std::cout << "No\n";
                return 0;
            } else {
                a[1]--;
            }
        }
        if (a[n - 1] == 2) {
            a[n - 1] = 0;
            if (a[n - 2] - 1 < 0) {
                std::cout << "No\n";
                return 0;
            } else {
                a[n - 2]--;
            }
        }
        int start = 0;
        int finish = n - 2;
        if (a[0] == 1) {
            for (int i = 0; i < n; i++) {
                if (a[i] == 0) {
                    start = i;
                    break;
                }
            }
        }
        if (start == n) {
            std::cout << "Yes\n";
            return 0;
        }
        if (a[n - 1] == 1) {
            for (int i = n - 1; i >= 0; i--) {
                if (a[i] == 0) {
                    finish = i;
                    break;
                }
            }
        }
        for (int i = start; i <= finish; i++) {
            if (a[i] == 1) {
                if (a[i + 1] == 0) {
                    std::cout << "No\n";
                    return 0;
                }
                a[i] = a[i + 1] = 0;
            }
```

```
        }
        std::cout << "Yes\n";
    }

    return 0;
}
```

Решение L

```
#include<bits/stdc++.h>
#define ll long long
#define MOD 1000000007

using namespace std;

template <class T>
inline void hash_combine(std::size_t & seed, const T & v)
{
std::hash<T> hasher;
seed ^= hasher(v) + 0x9e3779b9 + (seed << 6) + (seed >> 2);
}

namespace std
{
template<typename S, typename T> struct hash<pair<S, T>>
{
inline size_t operator()(const pair<S, T> & v) const
{
size_t seed = 0;
::hash_combine(seed, v.first);
::hash_combine(seed, v.second);
return seed;
}
};
}

struct crate {
int i;
int j;
};

int main() {
std::ios::sync_with_stdio(false);
int n;
cin >> n;
vector<crate> crates;
std::unordered_map <std::pair<ll, ll>, ll> mp;

for (int i = 0; i < n; ++i) {
int l, r;
```

```
std::cin >> l >> r;
mp[{l, r}] = 0;
crates.push_back({l, r});
}
sort(crates.begin(), crates.end(), [] (const crate& lhs, const crate& rhs) {
return lhs.i + lhs.j > rhs.i + rhs.j;
});
for (int i = 0; i < crates.size(); ++i) {
const crate& cr = crates[i];
ll& cur = mp[{cr.i, cr.j}];
if (!mp.count({cr.i + 1, cr.j}) && !mp.count({cr.i, cr.j + 1})) {
cur = 1;
}
if (mp.count({cr.i + 1, cr.j})) {
cur += mp[{cr.i + 1, cr.j}];
cur %= MOD;
}
if (mp.count({cr.i, cr.j + 1})) {
cur += mp[{cr.i, cr.j + 1}];
cur %= MOD;
}
}
cout << mp[{1,1}] << "\n";
}
```

Решение M

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

#define ull unsigned long long

const ull MOD = static_cast<ull>(std::pow(10, 9)) + 7;

ull mod_bin_pow(ull a, ull n) {
    ull res = 1;
    while (n) {
        if (n & 1) {
            res = (res % MOD * a % MOD) % MOD;
            --n;
        } else {
            a = (a % MOD * a % MOD) % MOD;
            n >>= 1;
        }
    }

    return res;
}

int main() {
    std::ios::sync_with_stdio(false);
```

```
    std::cin.tie(nullptr);
    int q;
    std::cin >> q;

    for (int it = 0; it < q; it++) {
        ull i1, i2, j1, j2;
        std::cin >> i1 >> i2 >> j1 >> j2;
        ull first = (mod_bin_pow(2, i1) % MOD * (mod_bin_pow(2, i2 - i1 + 1) % MOD - 1)
             % MOD) % MOD;
        ull second = (mod_bin_pow(3, j1) % MOD * ((mod_bin_pow(3, j2 - j1 + 1) % MOD -
            1) % MOD * mod_bin_pow(2, MOD - 2) % MOD) % MOD) % MOD;

        std::cout << (first * second) % MOD << "\n";
    }

    return 0;
}
```

RuCode 3.0 25.04.2021

Решение A

```
#include <iostream>

using namespace std;

int main() {
long long n;
int L;
cin >> n >> L;
int curr_team;
int dist, team_one_score=0, team_two_score=0;
for (int i = 0; i < n; ++i) {
cin >> curr_team >> dist;
if(curr_team == 1){
if (dist == -1){
team_one_score+=1;
} else if (dist >= L){
team_one_score+=3;
} else {
team_one_score+=2;
}
} else {
if (dist == -1){
team_two_score+=1;
} else if (dist >= L){
team_two_score+=3;
} else {
team_two_score+=2;
```

```
}
}
}
std::cout << team_one_score << ":" << team_two_score << std::endl;
return 0;
}
```

Grand Prix of Urals 09.05.2021

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	Astrology	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	1999	OK		278
B	Build the String	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		64 M	64 M	65536 bytes	2000			
C	Count Min Ratio	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		64 M	64 M	65536 bytes	2000			
D	Diophantine Equation	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	2000			
E	Emperor's Palace	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		64 M	64 M	65536 bytes	2000			
F	Five-pointed Queries	**input.txt** or *standard input*	**output.txt** or *standard output*	5000 ms		64 M	64 M	65536 bytes	2000			
G	Gas and Minerals	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	2000			
I	Infection	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	2000			
J	Jumping Path	**input.txt** or *standard input*	**output.txt** or *standard output*	3000 ms		64 M	64 M	65536 bytes	2000			
L	Laser Beam	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	2000			
M	Mirror Brackets	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	2000			
N	Numbers	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	1997	OK		321
O	Overwhelming Vowels	**input.txt** or *standard input*	**output.txt** or *standard output*	2000 ms		64 M	64 M	65536 bytes	1998	Wrong answer	4	300

Решение A

```
#include <iostream>
#include <string>


using namespace std;

int main() {
    string zod;
    string month, day;
    bool flag = false;
    cin >> zod;
    for (int i = 0; i < 10; ++i) {
        if(i == 5 || i == 6){
            month.push_back(zod[i]);
        }
        if(i == 8 || i == 9){
            day.push_back(zod[i]);
        }
    }
    int a = stoi(month);
    int b = stoi(day);
    if (a == 3 && b > 20){
        cout << "Aries" << endl;
    }
    if (a == 3 && b < 21){
        cout << "Pisces" << endl;
    }
    if (a == 4 && b < 20){
```

```
29          cout << "Aries" << endl;
30      }
31      if (a == 4 && b > 19){
32          cout << "Taurus" << endl;
33      }
34      if (a == 5 && b < 21){
35          cout << "Taurus" << endl;
36      }
37      if (a == 5 && b > 20){
38          cout << "Gemini" << endl;
39      }
40      if (a == 6 && b < 21){
41          cout << "Gemini" << endl;
42      }
43      if (a == 6 && b > 20){
44          cout << "Cancer" << endl;
45      }
46      if (a == 7 && b < 23){
47          cout << "Cancer" << endl;
48      }
49      if (a == 7 && b > 22){
50          cout << "Leo" << endl;
51      }
52      if (a == 8 && b < 23){
53          cout << "Leo" << endl;
54      }
55      if (a == 8 && b > 22){
56          cout << "Virgo" << endl;
57      }
58      if (a == 9 && b < 23){
59          cout << "Virgo" << endl;
60      }
61      if (a == 9 && b > 22){
62          cout << "Libra" << endl;
63      }
64      if (a == 10 && b < 23){
65          cout << "Libra" << endl;
66      }
67      if (a == 10 && b > 22){
68          cout << "Scorpio" << endl;
69      }
70      if (a == 11 && b < 23){
71          cout << "Scorpio" << endl;
72      }
73      if (a == 11 && b > 22){
74          cout << "Sagittarius" << endl;
75      }
76      if (a == 12 && b < 22){
77          cout << "Sagittarius" << endl;
```

```
    }
    if (a == 12 && b > 21){
        cout << "Capricorn" << endl;
    }
    if (a == 1 && b < 20){
        cout << "Capricorn" << endl;
    }
    if (a == 1 && b > 19){
        cout << "Aquarius" << endl;
    }
    if (a == 2 && b < 19){
        cout << "Aquarius" << endl;
    }
    if (a == 2 && b > 18){
        cout << "Pisces" << endl;
    }
    return 0;
}
```

Решение N

```
low, high = map(int, input().split())

ans = 0

for i in range(low, high + 1):
  if (i == 1):
    ans += 10
  else:
    ans += int('9' * i) - int('1' + '0' * (i - 1)) + 1

print(ans)
```

Решение O (не засчитано)

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string a;
    int counter = 0;
    cin >> a;
    int b = a.length();
    for (int i = 0; i < a.length(); ++i) {
        if(a[i] == 'a' || a[i] == 'e' || a[i] == 'i' || a[i] == 'o' || a[i] == 'u'){
            counter++;
        }
    }
    b = (b/2 + 1) - counter;
```

```
17      if(b > 0){
18          cout << b << endl;
19      } else {
20          cout << '0' << endl;
21      }
22  }
```