

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	35	25	460

Task A ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin() , x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in" , "r" , stdin);
    //freopen("7.out" , "w" , stdout);
    ll k;
    cin >> k;
    ll sum = 0;
    for(int i = max(1LL , k - 100); i < k; i++) {
        sum += i;
        sum /= 10;
    }
    sum += k;
    cout << sum % 10;
    return 0;
}
```

Task B ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin() , x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
const ll N = 1e5 + 1;
ll tree[4 * N];
void upd(ll v , ll l , ll r , ll i , ll x) {
    if(l + 1 == r) {
        tree[v] = x;
        return;
    }
    ll m = (r + 1) / 2;
    if(m > i) upd(2 * v + 1 , l , m , i , x);
    else upd(2 * v + 2 , m , r , i , x);
    tree[v] = min(tree[2 * v + 1] , tree[2 * v + 2]);
}
void build(ll v , ll l , ll r) {
    if(l + 1 == r) {
        tree[v] = 1e9;
        return;
    }
    ll m = (r + 1) / 2;
    build(2 * v + 1 , l , m);
    build(2 * v + 2 , m , r);
    tree[v] = min(tree[2 * v + 1] , tree[2 * v + 2]);
}
ll ans(ll v , ll l , ll r , ll L , ll R) {
    if(l >= L && r <= R) {
        return tree[v];
    }
    if(l >= R || r <= L) {
        return 1e9;
    }
    ll m = (r + 1) / 2;
    return min(ans(2 * v + 1 , l , m , L , R) , ans(2 * v + 2 , m , r , L , R));
}
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in" , "r" , stdin);
    //freopen("7.out" , "w" , stdout);
    ll n , k;
    cin >> n >> k;
    string s;
    cin >> s;
    vector <vector <ll>> cnt(n + 1 , vector <ll> (26));
    for(int i = 0; i < n; i++) {
        cnt[i + 1] = cnt[i];
        cnt[i + 1][s[i] - 'a'] += 1;
    }
    vector <ll> dp(n + 1 , 1e9);
    dp[0] = 0;
    build(0 , 0 , N);
```

```

upd(0 , 0 , N , 0 , 0);
for(int i = 1; i <= n; i++) {
    ll l = 0 , r = k + 1;
    while(r - l > 1) {
        ll mid = (r + l) / 2;
        if(i - mid + 1 < 1) r = mid;
        else {
            ll c = 0;
            for(int j = 0; j < 26; j++) {
                c += cnt[i][j] != cnt[i - mid][j];
            }
            if(c <= 3) l = mid;
            else r = mid;
        }
    }
    //cout << l << '\n';
    dp[i] = ans(0 , 0 , N , i - 1 , i) + 1;
    //cout << dp[i] << '\n';
    upd(0 , 0 , N , i , dp[i]);
}
cout << dp[n];
return 0;
}

```

Task C ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin() , x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in" , "r" , stdin);
    //freopen("7.out" , "w" , stdout);
    int n , x , y , sum1 = 0 , sum2 = 0;
    cin >> n >> x >> y;
    vector<int> v(n) , w(n);
    for(auto &i : v) {
        cin >> i;
        sum1 += i;
    }
    for(auto &i : w) {
        cin >> i;
        sum2 += i;
    }
    vector<vector<int>> dp(n + 1 , vector<int>(x + 1));
    for(int i = 0; i < n; i++) {
        for(int j = 0; j <= x; j++) {
            dp[i + 1][j] = dp[i][j];
            if(j - v[i] >= 0) {
                dp[i + 1][j] = max(dp[i + 1][j] , dp[i][j - v[i]] + w[i]);
            }
        }
    }
    if(sum2 - dp[n][x] > y) {
        cout << -1;
    } else {
        string ans;
        for(int i = 0; i < n; i++) {
            ans += 'y';
        }
        ll i = n , j = x;
        while(1) {
            if(dp[i][j] == 0) break;
            if(dp[i - 1][j] == dp[i][j]) {
                i--;
                continue;
            }
            if(dp[i - 1][j - v[i - 1]] + w[i - 1] == dp[i][j]) {
                i--, j -= v[i] , ans[i] = 'x';
            }
        }
        cout << ans;
    }
    return 0;
}
```

Task D ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin(), x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in", "r", stdin);
    //freopen("7.out", "w", stdout);
    ll n;
    cin >> n;
    string s;
    cin >> s;
    string p = "";
    for(auto x : s) {
        if(p.empty()) {
            if(x == '(' || x == ')') p += '(';
            else p += '[';
        } else {
            if((x == ')') || x == '(' && p.back() == '(') {
                p.pop_back();
            } else {
                if((x == ']' || x == '[') && p.back() == '[') {
                    p.pop_back();
                } else {
                    if(x == '(' || x == ')') p += '(';
                    else p += '[';
                }
            }
        }
    }
    stack<char> dlt;
    ll ans = 0;
    for(auto x : p) {
        if(dlt.empty()) {
            if(x == '(' || x == ')') dlt.push('(');
            else dlt.push('[');
        } else {
            if(x == ')') || x == '(' {
                if(dlt.top() != '(') {
                    ans++;
                }
                dlt.pop();
            } else {
                if(dlt.top() != '[') {
                    ans++;
                }
                dlt.pop();
            }
        }
    }
    cout << ans;
    return 0;
}
```

}

Task E ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin() , x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
vector <pair <vector <ll> , ll> > in , out;
set <vector <ll> > has;
vector <ll> now(3);
void solve(ll n) {
    if(n == 3) {
        for(int i = 1; i <= 10; i++) {
            if(now[0] != i && now[1] != i && now[2] != i) {
                vector <ll> pt = now;
                pt.push_back(i);
                sort(all(pt));
                if(has.find(pt) == has.end()) {
                    in.push_back({now , i});
                    has.insert(pt);
                    out.push_back({pt , i});
                    break;
                }
            }
        }
        return;
    }
    if(n == 0) {
        for(int i = 1; i <= 10; i++) {
            now[n] = i;
            solve(n + 1);
            now[n] = 0;
        }
    } else {
        for(int i = now[n - 1] + 1; i <= 10; i++) {
            now[n] = i;
            solve(n + 1);
            now[n] = 0;
        }
    }
}
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in" , "r" , stdin);
    //freopen("7.out" , "w" , stdout);
    solve(0);
    string type;
    cin >> type;
    if(type == "add") {
        ll t;
        cin >> t;
        while(t--) {
            ll n , k;
            cin >> n >> k;
            vector <ll> p(k);
```

```

        for(auto &x : p) cin >> x;
        sort(all(p));
        for(auto x : in) {
            if(x.first == p) cout << x.second;
        }
        cout << '\n';
    }
} else {
    ll t;
    cin >> t;
    while(t--) {
        ll n , k;
        cin >> n >> k;
        vector <ll> p(k + 1);
        for(auto &x : p) cin >> x;
        sort(all(p));
        for(auto x : out) {
            if(x.first == p) {
                for(auto i : p) if(i != x.second) cout << i << '␣';
            }
        }
        cout << '\n';
    }
}
return 0;
}

```


Task F ()

```
#include <iostream>
#include <map>
#include <cmath>
#include <algorithm>
#include <set>
#include <vector>
#include <queue>
#include <stack>
#include <istream>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <time.h>
#include <chrono>
#include <cstdio>
#include <fstream>
using namespace std;
#define ll long long
#define ld long double
#define all(x) x.begin() , x.end()
#define deb(z) cout << #z << ' ' << z << '\n';
#define pb push_back
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //freopen("4.in" , "r" , stdin);
    //freopen("7.out" , "w" , stdout);
    ll n;
    cin >> n;
    if(n < 9) {
        cout << 4 << '\n';
        cout << "0_0\n0_4\n4_4\n4_0\n";
        vector<string> ans = {"0_4" , "-4_4" , "-4_0" , "-4_-4" , "0_-4" , "4_-4" , "4_0" , "4_4"};
    };
    for(int i = 0; i < n; i++) cout << ans[i] << '\n';
}
return 0;
}
```