Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	80	30	52	0	362

Task A ()

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

using namespace std;

int main() {
        vector<int> a(6);
        vector<int> ans;

        for (int i = 0; i < 6; i++) {
                cin >> a[i];

                vector<int> last;

                for (int j = 0; j < i; j++) {
                        if (j + 1 == a[i]) {
                                last.push_back(i + 1);
                        }
                        last.push_back(ans[j]);
                }

                //cout << "AA";

                if (last.size() <= ans.size()) {
                        last.push_back(i + 1);
                }

                ans = last;

                //cout << "BB";

                /*for (auto elem : ans) {
                        cout << elem << " ";
                }
                cout << "\n";*/

        }

        for (int i = 1; i <= 6; i++) {
                for (int j = 0; j < 6; j++) {
                        if (i == ans[j]) {
                                cout << j + 1 << "␣";
                                break;
                        }
                }
        }

}
```

Task B ()

```cpp
#include <iostream>
#include <vector>
#include <deque>

using namespace std;

int main() {
	string s;
	cin >> s;

	int n;
	cin >> n;

	vector<int> a(n);

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

	if (s == "first") {
		long long sum = 0;

		for (auto elem : a) {
			sum += elem;
		}

		cout << sum * 100000 << "\n";

	}
	else {
		long long ans = 0;
		long long sum = a[0] / 100000;

		for (auto elem : a) {
			ans += elem % 100000;
		}

		cout << ans + sum << "\n";

	}

}

//1500000
```

Task C ()

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

using namespace std;

bool canFit(pair<int, int> a, pair<int, int> b) {
        return (b.first <= a.first && b.second <= a.second || b.second <= a.first && b.first <= a.
            second);
}

void merge(set<pair<int, int>>& a, set<pair<int, int>> b) {
        for (auto& elem : b) {
                a.insert(elem);
        }
}

set<pair<int, int>> twoBoxes(pair<int, int> a, pair<int, int> b) {
        set<pair<int, int>> ans;

        if (a.first == b.first && a.second != b.second) {
                ans.insert({ a.first, abs(a.second - b.second) });
        }
        if (a.first == b.second && a.second != b.first) {
                ans.insert({ a.first, abs(a.second - b.first) });
        }
        if (a.second == b.first && a.first != b.second) {
                ans.insert({ a.second, abs(a.first - b.second) });
        }
        if (a.second == b.second && b.first != a.first) {
                ans.insert({ a.second, abs(a.first - b.first) });
        }

        return ans;
}

set<pair<int, int>> threeBoxes(pair<int, int> a, pair<int, int> b, pair<int, int> c) {
        set<pair<int, int>> ans;

        ans.insert(a);
        ans.insert(b);
        ans.insert(c);

        merge(ans, twoBoxes(a, b));
        merge(ans, twoBoxes(b, c));
        merge(ans, twoBoxes(a, c));

        if (b.first == c.first) {
                merge(ans, twoBoxes(a, { b.first, b.second + c.second }));
        }
        if (b.second == c.second) {
                merge(ans, twoBoxes(a, { b.second, b.first + c.first }));
        }
        if (b.second == c.first) {
                merge(ans, twoBoxes(a, { b.second, b.first + c.second }));
        }
        if (b.first == c.second) {
                merge(ans, twoBoxes(a, { b.first, b.second + c.first }));
        }

        for (int l = 0; l < 2; l++) {
                for (int i = 0; i < 2; i++) {
                        for (int j = 0; j < 2; j++) {
                                for (int k = 0; k < 2; k++) {

                                        if (b.first < a.first && b.second < a.second && c.first +
                                            b.first >= a.first && c.second <= a.second - b.second)
                                             {
                                                ans.insert({ a.first - b.first, b.second });
                                        }
```

```
                                        swap(c.first, c.second);
                                }
                                swap(b.first, b.second);
                        }
                        swap(a.first, a.second);
                }
                swap(b, c);
        }

        return ans;
}

int main() {
        vector<pair<int, int>> b(3);

        for (int i = 0; i < 3; i++) {
                cin >> b[i].first >> b[i].second;
        }

        sort(b.begin(), b.end());
        reverse(b.begin(), b.end());

        //cout << "\n\n";

        set<pair<int, int>> ans;

        for (auto& elem : threeBoxes(b[0], b[1], b[2])) {
                if (elem.first > elem.second) {
                        ans.insert({ elem.second, elem.first });
                }
                else {
                        ans.insert({ elem.first, elem.second });
                }
        }

        for (auto& elem : ans) {
                cout << elem.first << " " << elem.second << "\n";
        }

}
```

Task D ()

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

using namespace std;

struct node {
	int vase, type, ans;
};

vector<int> v;
node d[2][2][2][51][51][51];

node solve(int i, int j, int k, int a, int b, int c) {
	if (d[i][j][k][a][b][c].ans == -1) {
		if (i == 1) {
			if (solve(0, j, k, v[0], b, c).ans == 0) {
				d[i][j][k][a][b][c] = { 1, 0, 1 };
			}
		}
		if (j == 1) {
			if (solve(i, 0, k, a, v[1], c).ans == 0) {
				d[i][j][k][a][b][c] = { 2, 0, 1 };
			}
		}
		if (k == 1) {
			if (solve(i, j, 0, a, b, v[2]).ans == 0) {
				d[i][j][k][a][b][c] = { 3, 0, 1 };
			}
		}

		for (int l = 1; l <= a; l++) {
			if (solve(i, j, k, a - l, b, c).ans == 0) {
				d[i][j][k][a][b][c] = { 1, l, 1 };
			}
		}

		for (int l = 1; l <= b; l++) {
			if (solve(i, j, k, a, b - l, c).ans == 0) {
				d[i][j][k][a][b][c] = { 2, l, 1 };
			}
		}

		for (int l = 1; l <= c; l++) {
			if (solve(i, j, k, a, b, c - l).ans == 0) {
				d[i][j][k][a][b][c] = { 3, l, 1 };
			}
		}

		if (d[i][j][k][a][b][c].ans == -1) {
			d[i][j][k][a][b][c] = { -1, -1, 0 };
		}
	}

	return d[i][j][k][a][b][c];
}

int main() {

	for (int i = 0; i < 2; i++) {
		for (int j = 0; j < 2; j++) {
			for (int k = 0; k < 2; k++) {
				for (int a = 0; a < 51; a++) {
					for (int b = 0; b < 51; b++) {
						for (int c = 0; c < 51; c++) {
							d[i][j][k][a][b][c] = { -1, -1, -1 };
						}
					}
				}
			}
```

```
        }
}

int n;
cin >> n;

v.resize(3, 0);

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

int i, j, k, a, b, c;

if (n == 1) {
        i = 1;
        j = 0;
        k = 0;
        a = v[0];
        b = 0;
        c = 0;
}
else if (n == 2) {
        i = 1;
        j = 1;
        k = 0;
        a = v[0];
        b = v[1];
        c = 0;
}
else {
        i = 1;
        j = 1;
        k = 1;
        a = v[0];
        b = v[1];
        c = v[2];
}

while (true) {
        node curMove = solve(i, j, k, a, b, c);

        if (curMove.ans == 0) {
                cout << "-1 -1\n";
                return 0;
        }
        else {
                cout << curMove.vase << " " << curMove.type << "\n";

                if (curMove.type == 0) {
                        if (curMove.vase == 1) {
                                a = v[0];
                                i = 0;
                        }
                        else if (curMove.vase == 2) {
                                b = v[1];
                                j = 0;
                        }
                        else {
                                c = v[2];
                                k = 0;
                        }
                }
                else {
                        if (curMove.vase == 1) {
                                a -= curMove.type;
                        }
                        else if (curMove.vase == 2) {
                                b -= curMove.type;
                        }
                        else {
                                c -= curMove.type;
                        }
                }
```

```
                }

                int ans1, ans2;
                cin >> ans1 >> ans2;

                if (ans1 == -1 && ans2 == -1) {
                        return 0;
                }
                else {
                        if (ans2 == 0) {
                                if (ans1 == 1) {
                                        a = v[0];
                                        i = 0;
                                }
                                else if (ans1 == 2) {
                                        b = v[1];
                                        j = 0;
                                }
                                else {
                                        c = v[2];
                                        k = 0;
                                }
                        }
                        else {
                                if (ans1 == 1) {
                                        a -= ans2;
                                }
                                else if (ans1 == 2) {
                                        b -= ans2;
                                }
                                else {
                                        c -= ans2;
                                }
                        }
                }

                cout.flush();
        }
}
```

Task E ()

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

using namespace std;

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

	string s;
	cin >> s;

	vector<int> d(100);
	d[0] = 1;

	for (int i = 1; i < 100; i++) {
		d[i] = 2 * d[i - 1];
	}

	while (t--) {
		if (s == "transmit") {
			int n;
			cin >> n;

			for (int i = 0; i < 10; i++) {
				if (n % 2 == 0) {
					for (int j = 0; j < 10; j++) {
						cout << "0";
					}
				}
				else {
					for (int j = 0; j <= i; j++) {
						cout << "1";
					}

					for (int j = i + 1; j < 10; j++) {
						cout << "0";
					}
				}

				n /= 2;
				cout << "\n";
			}
			cout << "\n";
		}
		else {

			int n = 0;

			for (int i = 0; i < 10; i++) {

				int count = 0;

				for (int j = 0; j < 10; j++) {
					char type;
					cin >> type;

					if (type == '1') {
						count++;
					}
				}
				if (count >= 1) {
					n += d[count - 1];
				}
			}

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

```
}
```

Task F ()

```
a = int(input())
b = int(input())

print(a + b)
```