

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. Шухова»
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Лабораторная работа №19.13
по дисциплине: «Множества»

Выполнил/а: ст. группы ВТ-231
Кисиль Николай Владимирович

Проверили:
Черников Сергей Викторович
Новожен Никита Викторович

Белгород, 2024 г.

**Цель работы: закрепление навыков работы со структурами,
изучение простых способов представления множеств в памяти ЭВМ**

Содержание работы

Задание 1: Выполнить реализацию множества на типе <code>uint32_t</code> . Содержимое файла <i>bitset.h</i> :	3
Задание 2: На неупорядоченном массиве:	3
Задание 3: *На упорядоченном массиве:	6
Задача 1. Определи маршрут (1056A)	9
Задача 2. Пропущенная серия (440A)	10
Задача 3. Перестановка букв (1093B)	11
Задача 4. Тихий класс (1166A)	12
Задача 5. Щедрый Кефа (841A)	13
Задача 6. Перекраска собачек (1025A)	14
Задача 7. Ступени (1011A)	15
Задача 8. Башни (37A)	16
Задача 9. Бейджик (1020B)	17
Задача 10. Разнообразие - это хорошо (672B)	18
Задача 11. Игра: Банковские карты (777B)	19
Задача 12. Противоположности притягиваются (131B)	20

Задание 1: Выполнить реализацию множества на типе uint32_t.

Содержимое файла *bitset.h*:

Файл *bitset.h* приложен к лабораторной работе

Задание 2: На неупорядоченном массиве:

Файл *unordered_array_set.h* приложен к лабораторной работе

Тесты:

```
#include "unordered_array_set.h"

void test_unordered_array_set_union1() {
    unordered_array_set set1 = unordered_array_set_create_from_array((int[])
{1, 2}, 2);
    unordered_array_set set2 = unordered_array_set_create_from_array((int[])
{2, 3}, 2);
    unordered_array_set resSet = unordered_array_set_union(set1, set2);
    unordered_array_set expectedSet =
unordered_array_set_create_from_array((int[]) {1, 2, 3}, 3);
    assert(unordered_array_set_isEqual(resSet, expectedSet));
    unordered_array_set_delete(set1);
    unordered_array_set_delete(set2);
    unordered_array_set_delete(resSet);
    unordered_array_set_delete(expectedSet);
}

void test_unordered_array_set_union2() {
    unordered_array_set set1 = unordered_array_set_create_from_array((int[])
{1, 2, 3}, 3);
    unordered_array_set set2 = unordered_array_set_create_from_array((int[])
{}, 0);
    unordered_array_set resSet = unordered_array_set_union(set1, set2);
    unordered_array_set expectedSet =
unordered_array_set_create_from_array((int[]) {1, 2, 3}, 3);
    assert(unordered_array_set_isEqual(resSet, expectedSet));
    unordered_array_set_delete(set1);
    unordered_array_set_delete(set2);
    unordered_array_set_delete(resSet);
    unordered_array_set_delete(expectedSet);
}

void test_unordered_array_set_union() {
    test_unordered_array_set_union1();
    test_unordered_array_set_union2();
}

void test_unordered_array_set_in1() {
    unordered_array_set set = {NULL, 0, 0};
    unordered_array_set_insert(&set, 5);
    unordered_array_set_insert(&set, 10);
    size_t position = unordered_array_set_in(set, 10);

    assert(position == 1);
    unordered_array_set_delete(set);
}

void test_unordered_array_set_in2() {
    unordered_array_set set = {NULL, 0, 0};
    unordered_array_set_insert(&set, 5);
```

```

        unordered_array_set_insert(&set, 10);
        size_t position = unordered_array_set_in(set, 15);

        assert(position == set.size);
        unordered_array_set_delete(set);
    }

    void test_unordered_array_set_in() {
        test_unordered_array_set_in1();
        test_unordered_array_set_in2();
    }

    void test_unordered_array_set_isSubset1() {
        unordered_array_set emptySet = {NULL, 0, 0};
        unordered_array_set set = {NULL, 0, 0};
        unordered_array_set_insert(&set, 5);

        assert(!unordered_array_set_isSubset(emptySet, set));
    }

    void test_unordered_array_set_isSubset2() {
        unordered_array_set set = {NULL, 0, 0};
        unordered_array_set_insert(&set, 5);
        unordered_array_set_insert(&set, 10);

        assert(unordered_array_set_isSubset(set, set));
    }

    void test_unordered_array_set_isSubset3() {
        unordered_array_set set1 = {NULL, 0, 0};
        unordered_array_set set2 = {NULL, 0, 0};
        unordered_array_set_insert(&set1, 5);
        unordered_array_set_insert(&set2, 10);

        assert(!unordered_array_set_isSubset(set1, set2));
    }

    void test_unordered_array_set_isSubset() {
        test_unordered_array_set_isSubset1();
        test_unordered_array_set_isSubset2();
        test_unordered_array_set_isSubset3();
    }

    void test_unordered_array_set_insert() {
        unordered_array_set set = {NULL, 0, 0};
        unordered_array_set_insert(&set, 5);

        assert(set.size == 1 && set.data[0] == 5);
        unordered_array_set_delete(set);
    }

    void test_unordered_array_set_deleteElement() {
        unordered_array_set set = {NULL, 0, 0};
        unordered_array_set_insert(&set, 5);
        unordered_array_set_insert(&set, 10);
        unordered_array_set_deleteElement(&set, 5);

        assert(set.size == 1 && set.data[0] == 10);

        unordered_array_set_delete(set);
    }

    void test_unordered_array_set_intersection() {

```

```

unordered_array_set set1 = {NULL, 0, 0};
unordered_array_set set2 = {NULL, 0, 0};
unordered_array_set insert(&set1, 5);
unordered_array_set insert(&set1, 10);
unordered_array_set insert(&set2, 10);
unordered_array_set insert(&set2, 15);
unordered_array_set intersection = unordered_array_set_intersection(set1,
set2);

assert(intersection.size == 1 && intersection.data[0] == 10);

unordered_array_set_delete(set1);
unordered_array_set_delete(set2);
unordered_array_set_delete(intersection);
}

void test_unordered_array_set_difference() {
    unordered_array_set set1 = {NULL, 0, 0};
    unordered_array_set set2 = {NULL, 0, 0};
    unordered_array_set insert(&set1, 5);
    unordered_array_set insert(&set1, 10);
    unordered_array_set insert(&set2, 10);
    unordered_array_set insert(&set2, 15);
    unordered_array_set difference = unordered_array_set_difference(set1,
set2);

    assert(difference.size == 1 && difference.data[0] == 5);

    unordered_array_set_delete(set1);
    unordered_array_set_delete(set2);
    unordered_array_set_delete(difference);
}

void test_unordered_array_set_symmetricDifference() {
    unordered_array_set set1 = {NULL, 0, 0};
    unordered_array_set set2 = {NULL, 0, 0};
    unordered_array_set insert(&set1, 5);
    unordered_array_set insert(&set1, 10);
    unordered_array_set insert(&set2, 10);
    unordered_array_set insert(&set2, 15);
    unordered_array_set symmetric_difference =
unordered_array_set_symmetricDifference(set1, set2);

    assert(symmetric_difference.size == 2 && symmetric_difference.data[0] ==
5 && symmetric_difference.data[1] == 15);

    unordered_array_set_delete(set1);
    unordered_array_set_delete(set2);
    unordered_array_set_delete(symmetric_difference);
}

void test_unordered_array_set_complement() {
    unordered_array_set set = {NULL, 0, 0};
    unordered_array_set insert(&set, 5);
    unordered_array_set insert(&set, 10);
    unordered_array_set universumSet = {NULL, 0, 0};
    unordered_array_set insert(&universumSet, 5);
    unordered_array_set insert(&universumSet, 10);
    unordered_array_set insert(&universumSet, 15);
    unordered_array_set complement = unordered_array_set_complement(set,
universumSet);

    assert(complement.size == 1 && complement.data[0] == 15);
}

```

```

        unordered_array_set_delete(set);
        unordered_array_set_delete(universumSet);
        unordered_array_set_delete(complement);
    }

void test() {
    test_unordered_array_set_in();
    test_unordered_array_set_isSubset();
    test_unordered_array_set_insert();
    test_unordered_array_set_union();
    test_unordered_array_set_deleteElement();
    test_unordered_array_set_intersection();
    test_unordered_array_set_difference();
    test_unordered_array_set_symmetricDifference();
    test_unordered_array_set_complement();
}

int main() {
    test();

    return 0;
}

```

Задание 3: *На упорядоченном массиве:

Файл `ordered_array_set.h` приложен к лабораторной работе

Тесты:

```

#include <stdio.h>
#include <assert.h>
#include "ordered_array_set.h"

void test_ordered_array_set_in() {
    ordered_array_set set = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4, 5}, 5);
    assert(ordered_array_set_in(&set, 3) == 2);
    assert(ordered_array_set_in(&set, 6) == set.size);
    ordered_array_set_delete(set);
}

void test_ordered_array_set_isSubset() {
    ordered_array_set set1 = ordered_array_set_create_from_array((int[]) {1,
2, 3}, 3);
    ordered_array_set set2 = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4, 5}, 5);
    ordered_array_set set3 = ordered_array_set_create_from_array((int[]) {1,
2, 6}, 3);
    assert(ordered_array_set_isSubset(set1, set2) == true);
    assert(ordered_array_set_isSubset(set2, set1) == false);
    assert(ordered_array_set_isSubset(set1, set3) == false);
    ordered_array_set_delete(set1);
    ordered_array_set_delete(set2);
    ordered_array_set_delete(set3);
}

void test_ordered_array_set_insert() {
    ordered_array_set set = ordered_array_set_create(5);
    ordered_array_set_insert(&set, 3);
    ordered_array_set_insert(&set, 1);
    ordered_array_set_insert(&set, 4);
}

```

```

        ordered_array_set_insert(&set, 2);
        ordered_array_set_insert(&set, 5);
        assert(ordered_array_set_in(&set, 3) == 2);
        assert(ordered_array_set_in(&set, 6) == set.size);
        ordered_array_set_delete(set);
    }

void test_ordered_array_set_union() {
    ordered_array_set set1 = ordered_array_set_create_from_array((int[]) {1,
2, 4}, 3);
    ordered_array_set set2 = ordered_array_set_create_from_array((int[]) {2,
3, 5}, 3);
    ordered_array_set result = ordered_array_set_union(set1, set2);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {1, 2, 3, 4, 5}, 5);
    assert(ordered_array_set_isEqual(result, expectedSet) == true);
    ordered_array_set_delete(set1);
    ordered_array_set_delete(set2);
    ordered_array_set_delete(result);
    ordered_array_set_delete(expectedSet);
}

void test_ordered_array_set_deleteElement() {
    ordered_array_set set = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4, 5}, 5);
    ordered_array_set_deleteElement(&set, 3);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {1, 2, 4, 5}, 4);
    assert(ordered_array_set_isEqual(set, expectedSet) == true);
    ordered_array_set_delete(set);
    ordered_array_set_delete(expectedSet);
}

void test_ordered_array_set_intersection() {
    ordered_array_set set1 = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4}, 4);
    ordered_array_set set2 = ordered_array_set_create_from_array((int[]) {2,
3, 5}, 3);
    ordered_array_set result = ordered_array_set_intersection(set1, set2);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {2, 3}, 2);
    assert(ordered_array_set_isEqual(result, expectedSet) == true);
    ordered_array_set_delete(set1);
    ordered_array_set_delete(set2);
    ordered_array_set_delete(result);
    ordered_array_set_delete(expectedSet);
}

void test_ordered_array_set_difference() {
    ordered_array_set set1 = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4}, 4);
    ordered_array_set set2 = ordered_array_set_create_from_array((int[]) {2,
3, 5}, 3);
    ordered_array_set result = ordered_array_set_difference(set1, set2);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {1, 4}, 2);
    assert(ordered_array_set_isEqual(result, expectedSet) == true);
    ordered_array_set_delete(set1);
    ordered_array_set_delete(set2);
    ordered_array_set_delete(result);
    ordered_array_set_delete(expectedSet);
}

```

```

void test_ordered_array_set_symmetricDifference() {
    ordered_array_set set1 = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4}, 4);
    ordered_array_set set2 = ordered_array_set_create_from_array((int[]) {2,
3, 5}, 3);
    ordered_array_set result = ordered_array_set_symmetricDifference(set1,
set2);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {1, 4, 5}, 3);
    assert(ordered_array_set_isEqual(result, expectedSet) == true);
    ordered_array_set_delete(set1);
    ordered_array_set_delete(set2);
    ordered_array_set_delete(result);
    ordered_array_set_delete(expectedSet);
}

void test_ordered_array_set_complement() {
    ordered_array_set set = ordered_array_set_create_from_array((int[]) {1,
2, 3, 4, 5}, 5);
    ordered_array_set universumSet =
ordered_array_set_create_from_array((int[]) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
10);
    ordered_array_set result = ordered_array_set_complement(set,
universumSet);
    ordered_array_set expectedSet =
ordered_array_set_create_from_array((int[]) {6, 7, 8, 9, 10}, 5);
    assert(ordered_array_set_isEqual(result, expectedSet) == true);
    ordered_array_set_delete(set);
    ordered_array_set_delete(universumSet);
    ordered_array_set_delete(result);
    ordered_array_set_delete(expectedSet);
}

void test() {
    test_ordered_array_set_in();
    test_ordered_array_set_isSubset();
    test_ordered_array_set_insert();
    test_ordered_array_set_union();
    test_ordered_array_set_deleteElement();
    test_ordered_array_set_intersection();
    test_ordered_array_set_difference();
    test_ordered_array_set_symmetricDifference();
    test_ordered_array_set_complement();
}

int main() {
    test();

    return 0;
}

```


Задача 1. Определи маршрут (1056A)

Код:

```
#include <stdio.h>
#include "ordered_array_set.h"

#define MAX_STOPS 101

int main() {
    int n;
    scanf("%d", &n);

    ordered_array_set route = ordered_array_set_create(MAX_STOPS);
    for (size_t i = 0; i < MAX_STOPS; i++) {
        route.data[i] = 0;
    }

    for (int i = 0; i < n; i++) {
        int r;
        scanf("%d", &r);

        for (int j = 0; j < r; j++) {
            int num_route;
            scanf("%d", &num_route);

            route.data[num_route]++;
        }
    }

    for (int i = 1; i < MAX_STOPS; i++) {
        if (route.data[i] == n) {
            printf("%d ", i);
        }
    }

    return 0;
}
```

Вердикт тестирующей системы:

246819419	17.02.2024 17:26	n1kuso	1056A - Определи маршрут	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	------------------	--------	--	---------	----------------	-------	--------

Задача 2. Пропущенная серия (440A)

Код:

```
#include <stdio.h>
#include "ordered_array_set.h"

int main() {
    int n;
    scanf("%d", &n);

    ordered_array_set episodes = ordered_array_set_create(n - 1);
    int viewed_episodes = 0;
    for(int i = 1; i < n; i++) {
        scanf("%d", &episodes.data[i]);
        viewed_episodes += episodes.data[i];
    }

    int total_episodes = 0;
    for (int i = 1; i <= n; i++) {
        total_episodes += i;
    }

    int missed_episode = total_episodes - viewed_episodes;

    printf("%d", missed_episode);

    return 0;
}
```

Вердикт тестирующей системы:

246280556	13.02.2024 22:37	n1kuso	A - Пропущенная серия	GNU C11	Полное решение	31 мс	300 КБ
---------------------------	---------------------	--------	---	------------	---------------------------	-------	--------

Задача 3. Перестановка букв (1093B)

Код:

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

#define SWAP(type, a, b) { \
    type t = a; \
    a = b; \
    b = t; \
}

typedef unsigned long long ull;

int main() {
    int sets;
    scanf("%d", &sets);

    for (int t_sets = 0; t_sets < sets; t_sets++) {
        char c[1001];
        scanf("%s", c);

        ull length = strlen(c);
        bool flag = false;

        for(size_t i = 0; i < length; i++) {
            if(c[i] != c[0]) {
                SWAP(char, c[i], c[length - 1]);
                flag = true;
                break;
            }
        }

        if(!flag) {
            printf("-1\n");
        } else {
            printf("%s\n", c);
        }
    }

    return 0;
}
```

Вердикт тестирующей системы:

246835079	17.02.2024 17:46	n1kuso	В - Перестановка букв	GNU C11	Полное решение	46 мс	300 КБ
---------------------------	---------------------	--------	---	------------	-------------------	-------	--------

Задача 4. Тихий класс (1166A)

Код:

```
#include <stdio.h>
#include "ordered_array_set.h"

#define COUNT_LETTERS 26

int main() {
    int n;
    scanf("%d", &n);

    ordered_array_set class = ordered_array_set_create(COUNT_LETTERS);
    for (size_t i = 0; i < COUNT_LETTERS; i++) {
        class.data[i] = 0;
    }
    getchar();

    char name[21];
    for (int i = 0; i < n; i++) {
        scanf("%s", name);
        class.data[name[0] - 'a']++;
    }

    int result = 0;
    for (int i = 0; i < COUNT_LETTERS; i++) {
        int first_pair = class.data[i] / 2;
        int second_pair = class.data[i] - first_pair;

        first_pair = first_pair * (first_pair - 1) / 2;
        second_pair = second_pair * (second_pair - 1) / 2;

        result += first_pair + second_pair;
    }

    printf("%d\n", result);

    return 0;
}
```

Вердикт тестирующей системы:

246817669	17.02.2024 17:11	n1kuso	А - Тихий класс	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	---------------------	--------	---------------------------------	------------	-------------------	-------	--------

Задача 5. Щедрый Кефа (841A)

Код:

```
#include <stdio.h>
#include "ordered_array_set.h"

#define COUNT_LETTERS 26

int main() {
    int n;
    scanf("%d", &n);

    ordered_array_set class = ordered_array_set_create(COUNT_LETTERS);
    for (size_t i = 0; i < COUNT_LETTERS; i++) {
        class.data[i] = 0;
    }
    getchar();

    char name[21];
    for (int i = 0; i < n; i++) {
        scanf("%s", name);
        class.data[name[0] - 'a']++;
    }

    int result = 0;
    for (int i = 0; i < COUNT_LETTERS; i++) {
        int first_pair = class.data[i] / 2;
        int second_pair = class.data[i] - first_pair;

        first_pair = first_pair * (first_pair - 1) / 2;
        second_pair = second_pair * (second_pair - 1) / 2;

        result += first_pair + second_pair;
    }

    printf("%d\n", result);

    return 0;
}
```

Вердикт тестирующей системы:

246841841	17.02.2024 17:53	n1kuso	А - Щедрый Кефа	GNU C11	Полное решение	31 мс	300 КБ
---------------------------	---------------------	--------	---------------------------------	------------	-------------------	-------	--------

Задача 6. Перекраска собачек (1025A)

Код:

```
#include <stdio.h>
#include <stdbool.h>
#include "ordered_array_set.h"

#define COUNT_LETTERS 26

int main() {
    int n;
    scanf("%d", &n);

    ordered_array_set count_colors = ordered_array_set_create(COUNT_LETTERS);
    for (size_t i = 0; i < COUNT_LETTERS; i++) {
        count_colors.data[i] = 0;
    }

    char colors[n];
    scanf("%s", colors);

    for (int i = 0; i < n; i++) {
        count_colors.data[colors[i] - 'a']++;
    }

    bool flag = false;

    for(size_t i = 0; i < COUNT_LETTERS; i++) {
        if(n == 1 || count_colors.data[i] > 1) {
            flag = true;
            break;
        }
    }

    if (flag)
        printf("YES");
    else
        printf("NO");

    return 0;
}
```

Вердикт тестирующей системы:

246543269	15.02.2024 18:44	n1kuso	A - Перекраска собачек	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	---------------------	--------	--	------------	-------------------	-------	--------

Задача 7. Ступени (1011A)

Код:

```
#include <stdio.h>
#include "ordered_array_set.h"

#define COUNT_LETTERS 26

int main() {
    int n, k;
    scanf("%d %d", &n, &k);

    ordered_array_set mass = ordered_array_set_create(COUNT_LETTERS);
    for (size_t i = 0; i < COUNT_LETTERS; i++) {
        mass.data[i] = 0;
    }

    char weights[n];
    scanf("%s", weights);

    for (int i = 0; i < n; i++) {
        mass.data[weights[i] - 'a']++;
    }

    int min_weight = 0;
    for (size_t i = 0; i < COUNT_LETTERS; i++) {
        if (mass.data[i] > 0 && k > 0) {
            i++;
            k--;
            min_weight += i;
        }
    }

    if (k == 0) {
        printf("%d\n", min_weight);
    } else {
        printf("%d\n", -1);
    }

    return 0;
}
```

Вердикт тестирующей системы:

246848483	17.02.2024 18:03	n1kuso	А - Ступени	GNU C11	Полное решение	30 мс	300 КБ
---------------------------	---------------------	--------	-----------------------------	------------	-------------------	-------	--------

Задача 8. Башни (37A)

Код:

```
#include <stdio.h>

#define MAX_NUMBER_STUDENTS 1001

int main() {
    int number_students;
    scanf("%d", &number_students);

    int array[number_students];

    for (int i = 1; i <= number_students; i++) {
        scanf("%d", &array[i]);
    }

    int holes_in_badges[MAX_NUMBER_STUDENTS] = {0};

    for (int i = 1; i <= number_students; i++) {
        int j = i;
        while (holes_in_badges[j] != i) {
            holes_in_badges[j] = i;
            j = array[j];
        }

        printf("%d ", j);
    }

    return 0;
}
```

Вердикт тестирующей системы:

246851819	17.02.2024 18:09	n1kuso	А - Башни	GNU C11	Полное решение	60 мс	300 КБ
---------------------------	---------------------	--------	---------------------------	------------	-------------------	-------	--------

Задача 9. Бейджик (1020B)

Код:

```
#include <stdio.h>

#define MAX_STUDENTS 1001

int main() {
    int n;
    scanf("%d", &n);

    int array[n];

    for (int i = 1; i <= n; i++) {
        scanf("%d", &array[i]);
    }

    int holes[MAX_STUDENTS] = {0};

    for (int i = 1; i <= n; i++) {
        int j = i;
        while (holes[j] != i) {
            holes[j] = i;
            j = array[j];
        }

        printf("%d ", j);
    }

    return 0;
}
```

Вердикт тестирующей системы:

246877737	17.02.2024 19:17	n1kuso	В - Бейджик	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	---------------------	--------	-----------------------------	------------	-------------------	-------	--------

Задача 10. Разнообразие - это хорошо (672B)

Код:

```
#include <stdio.h>

#define COUNT_LETTERS 26

int main() {
    int n;
    scanf("%d", &n);

    char s[n + 1];
    scanf("%s", s);

    int count = 0;

    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            if (s[i] == s[j]) {
                count++;
                break;
            }
        }
    }

    if (n > COUNT_LETTERS) {
        printf("-1");
    } else {
        printf("%d", count);
    }

    return 0;
}
```

Вердикт тестирующей системы:

246878679	17.02.2024 19:20	n1kuso	В - Разнообразие - это хорошо	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	---------------------	--------	---	------------	-------------------	-------	--------

Задача 11. Игра: Банковские карты (777В)

Код:

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_NUMBERS 1001

int compare(const void *p1, const void *p2) {
    return *((char *) p1) - *((char *) p2);
}

int main() {
    int n;
    scanf("%d\n", &n);

    char sherlock_card[MAX_NUMBERS];
    char moriarty_card[MAX_NUMBERS];

    scanf("%s\n %s", sherlock_card, moriarty_card);

    qsort(sherlock_card, n, sizeof(char), compare);
    qsort(moriarty_card, n, sizeof(char), compare);

    int number_clicks = 0;

    for (int i = 0, j = 0; i < n; i++) {
        while (j < n && sherlock_card[i] > moriarty_card[j]) {
            j++;
        }
        if (j >= n) {
            number_clicks++;
        }
        j++;
    }

    printf("%d\n", number_clicks);

    number_clicks = 0;

    for (int i = 0, j = 0; i < n; i++) {
        while (j < n && sherlock_card[i] >= moriarty_card[j]) {
            j++;
        }
        if (j < n) {
            number_clicks++;
        }
        j++;
    }

    printf("%d\n", number_clicks);

    return 0;
}
```

Вердикт тестирующей системы:

246882775	17.02.2024 19:36	n1kuso	В - Игра: Банковские карты	GNU C11	Полное решение	15 мс	300 КБ
---------------------------	---------------------	--------	--	---------	----------------	-------	--------

Задача 12. Противоположности притягиваются (131B)

Код:

```
#include <stdio.h>

#define MAX_NUM 25

int main() {
    int n;
    scanf("%d", &n);

    long long arr[MAX_NUM] = {0};

    int count = 0;
    long long sum = 0;

    for (int i = 0; i < n; i++) {
        int t;
        scanf("%d", &t);

        if (t == 0) {
            sum += count;
            count++;
        } else {
            arr[t + 10]++;
        }
    }

    for (int i = 0; i < 10; i++) {
        sum = sum + arr[i] * arr[20 - i];
    }

    printf("%lld", sum);

    return 0;
}
```

Вердикт тестирующей системы:

246884498	17.02.2024 19:42	n1kuso	В - Противоположности притягиваются	GNU C11	Полное решение	62 мс	300 КБ
---------------------------	---------------------	--------	---	------------	-------------------	-------	--------

Вывод: закрепили навыки работы со структурами, изучили простые способы представления множеств в памяти ЭВМ