



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ** | **Факультет прикладной
математики и информатики**

Кафедра прикладной математики
Практическое задание № 4
по дисциплине «Структуры данных и алгоритмы»

УПРАВЛЕНИЕ ТАБЛИЦАМИ

Бригада 2	ТАДЖИБАЕВ ЗАВКИДДИН
Группа ПМ-25	ЯГОДКИН АЛЕКСНАДР
Вариант 3	СУХАРЕВА СОФЬЯ

Преподаватель ТРАКИМУС ЮРИЙ ВИКТОРОВИЧ

Новосибирск, 2023

1 Задание

В спортивных соревнованиях участвуют n команд. В файле SPORT содержатся прогнозы результатов соревнований. Каждый прогноз включает номер команды, занявшей первое место, номер команды, занявшей последнее место, номера команд, входящих в первую тройку сильнейших команд. Построить таблицу, содержащую проценты голосов, отданных командам – претендентам на первое место, командам – претендентам на последнее место и проценты голосов, отданных командам – претендентам на первую тройку. Организовать таблицу как:

- а) неупорядоченную;
- б) упорядоченную;
- в) таблицу с вычисляемым входом.

2 Анализ программы

Входные данные: В файл SPORT.txt вводятся прогнозы результатов соревнований, каждый прогноз включает номер команды, занявшей первое место, номер команды, занявшей последнее место, номера команд, входящих в первую тройку сильнейших команд.

Выходные данные: В файл disordered.txt выводится таблица в порядке ввода данных в файл SPORT.txt. В файл ordered.txt выводится упорядоченная таблица. В файл calculated_input.txt выводится таблица с вычисляемым входом. “Файл SPORT.txt пуст.”; “Не удалось открыть файл SPORT.txt”; “ Не удалось открыть файл ordered.txt ”; “ Не удалось открыть файл disordered.txt”; “ Не удалось открыть файл calculated_input.txt”;

Анализ программы: После ввода данных в SPORT.txt программа обрабатывает данные и формирует таблицы с результатами побед на первом месте, последнем месте и топ-3. Поскольку могут возникнуть коллизии используется метод их разрешения. Создаются хеш-таблица в виде связанного списка, который содержит все элементы, хеш-коды которых соответствует определенному индексу в таблице. При добавлении нового элемента в хеш-таблицу сначала вычисляется его хеш-код. Затем этот элемент добавляется в связанный список, соответствующий данному индексу в таблице. Если в этом списке уже есть элемент с таким же ключом, то он будет добавлен в список, как новый узел, что позволяет хранить несколько элементов с одним и тем же ключом. При поиске элемента по ключу также вычисляется его хеш-код и затем производится поиск по связанному списку, соответствующему данному индексу в таблице. Программа формирует таблицу disordered.txt в том порядке, в котором элементы были заданы в SPORT.txt, происходит это при помощи функции table_static() в которой создается таблица и добавляются элементы в нее в порядке в котором они были заданы. В ordered.txt программа формирует таблицу при помощи функции binary_search(), что позволяет получить на выходе уже отсортированную таблицу. Файл calculated_input.txt содержит таблицу с вычисляемым входом, данные которой вводятся из консоли.

3 Программа

```
#include <stdio.h>
#include <windows.h>
#include <locale.h>

const UINT N = 1000, M = 20;
UINT K = 0;

struct elem
{
    UINT num, first, last, top3;
    elem(UINT _num = 0) :
```

```

        num(_num), first(0), last(0), top3(0) { };
};

struct table_static
{
    elem *el[N]{ };
    UINT size = 0;

    UINT insert(UINT x)
    {
        if (size >= N)
            return 0;

        for (UINT i = 0; i < size; i++)
            if (el[i].num == x)
                return i;

        el[size]->num = x;
        ++size;
        return size - 1;
    }

    UINT binary_search(UINT x)
    {
        int l = 0, r = size, m = 0;

        for ( ; l <= r; )
        {
            m = (l + r) / 2;
            if (el[m]->num == x)
                return m;
            el[m]->num < x ? l = m + 1 : r = m - 1;
        }
        return el[m]->num > x ? m : m + 1;
    }
} T1, T2;

struct list
{
    elem data{ };
    list *next = NULL;

    list(list *_next = NULL, UINT _num = 0) :
        next(_next), data({ _num }) { };
};

UINT hash(UINT x) { return x % M; }

struct table_dynamic
{

```

```

list *elm = NULL;

void insert(UINT a, UINT i)
{
    list *&p = elm, *s = p;
    for ( ; s && s->data.num != a; s = s->next) p = s;
    if (!s)
    {
        s = new list(NULL, a);
        p ? p->next = s : p = s;
    }
    switch (i)
    {
        case 0: s->data.first++; break;
        case 1: s->data.last++; break;
        default: s->data.top3++; break;
    }
    p = elm;
}
} T[M]{ };

bool input()
{
    FILE *f = NULL;
    UINT a = 0, i = 0, j = 0;
    T2.el[0]->num = UINT_MAX;

    fopen_s(&f, "SPORT.txt", "r");
    if (f)
    {
        for ( ; fscanf_s(f, "%d ", &a) == 1; K++)
        {
            i = T1.insert(a);
            j = T2.binary_search(a);

            if (T2.el[j]->num != a)
            {
                T2.size++;
                for (UINT k = T2.size; k > j; --k)
                    T2.el[k] = T2.el[k - 1];

                T2.el[j] = new elem(a);
            }

            switch (K % 5)
            {
                case 0: T1.el[i]->first++; T2.el[j]->first++; break;
                case 1: T1.el[i]->last++; T2.el[j]->last++; break;
                default: T1.el[i]->top3++; T2.el[j]->top3++; break;
            }
        }
    }
}

```

```

        K /= 5;
        fclose(f);
        return true;
    }
    else
        return false;
}

void output(FILE *&f, table_static t)
{
    fprintf_s(f, "№          First, %          Last, %          Top3, %\n");

    for (UINT i = 0; i < t.size; i++)
        fprintf_s(f, "%4.d %9.1f %9.1f %9.1f\n", t.el[i]->num,
            t.el[i]->first * 100. / K, t.el[i]->last * 100. / K,
            t.el[i]->top3 * 100. / K);
}

void output_calculated_input()
{
    FILE *f = NULL;

    fopen_s(&f, "calculated_input.txt", "w");
    if (f)
    {
        fprintf_s(f, "№          First, %          Last, %          Top3, %\n");

        for (UINT i = 0; i < M; i++)
            for (list *e = T[i].elm; e; e = e->next)
                fprintf_s(f, "%4.d %9.1 f %9.1f %9.1f\n", e->data.num,
                    e->data.first * 100. / K, e->data.last * 100. / K,
                    e->data.top3 * 100. / K);

        fclose(f);
    }
    else
        printf_s("Не удалось открыть файл calculated_input.txt.");
}

int main()
{
    setlocale(0, "");
    UINT cp = GetConsoleCP(), outcp = GetConsoleOutputCP();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    UINT a = 0;

    if (input())
    {
        if (K > 0)
        {

```

```

FILE *f1 = NULL, *f2 = NULL;

fopen_s(&f1, "disordered.txt", "w");
if (f1)
    output(f1, T1);
else
    printf_s("Не удалось открыть файл disordered.txt.");

fopen_s(&f2, "ordered.txt", "w");
if (f2)
    output(f2, T2);
else
    printf_s("Не удалось открыть файл ordered.txt.");

K = 0;
printf_s("Вводите прогноз: \n");
scanf_s("%d", &a);

for (; a; K++)
{
    T[hash(a)].insert(a, K % 5);
    scanf_s("%d", &a);
}

K /= 5;
output_calculated_input();
}
else
    printf_s("Файл SPORT.txt пуст.");
}
else
    printf_s("Не удалось открыть файл SPORT.txt.");
SetConsoleCP(cp);
SetConsoleOutputCP(outcp);
return 0;
}

```

4 Набор тестов

№	Входные данные	Назначение
1		Файл SPORT.txt отсутствует. Файл disordered.txt отсутствует. файл calculated_input.txt отсутствует. Файл calculated_input.txt отсутствует.
2		Файл SPORT.txt пуст.
3	1 2 1 6 8 9 5 9 8 6 5 6 5 8 7 5 3 6 5 4 Данные для calculated_input: 6 7 8 9 19 22 11 2 3 4 2 3 4 5 1 2 3 5 9 11	Обработка информации о результативности спортивных команд, добавление новых прогнозов
4	1 2 3 4 5 34 4 6 7 8 7 5 6 23 4 4 45 6 7 9 3 4 5 6 9 1 45 7 8 3 6 7 8 23 43 Данные для calculated_input: 6 7 10 15 20 6 7 10 15 20 13 14 1 2 3 7 6 8 9 19	Обработка информации о результативности спортивных команд, добавление голосов к старым прогнозам

5 Результаты работы программы

№	Ввод/Вывод программы																																								
1	Не удалось открыть файл SPORT.txt. Не удалось открыть файл disordered.txt. Не удалось открыть файл calculated_input.txt. Не удалось открыть файл ordered.txt.																																								
2	Файл SPORT.txt. пуст.																																								
3	disordered.txt <table><tr><th>№</th><th>First, %</th><th>Last, %</th><th>Top3, %</th></tr><tr><td>1</td><td>25,0</td><td>0,0</td><td>25,0</td></tr><tr><td>2</td><td>0,0</td><td>25,0</td><td>0,0</td></tr><tr><td>6</td><td>0,0</td><td>25,0</td><td>75,0</td></tr><tr><td>8</td><td>0,0</td><td>0,0</td><td>75,0</td></tr><tr><td>9</td><td>25,0</td><td>0,0</td><td>25,0</td></tr><tr><td>5</td><td>50,0</td><td>25,0</td><td>50,0</td></tr><tr><td>7</td><td>0,0</td><td>0,0</td><td>25,0</td></tr><tr><td>3</td><td>0,0</td><td>25,0</td><td>0,0</td></tr><tr><td>4</td><td>0,0</td><td>0,0</td><td>25,0</td></tr></table> ordered.txt	№	First, %	Last, %	Top3, %	1	25,0	0,0	25,0	2	0,0	25,0	0,0	6	0,0	25,0	75,0	8	0,0	0,0	75,0	9	25,0	0,0	25,0	5	50,0	25,0	50,0	7	0,0	0,0	25,0	3	0,0	25,0	0,0	4	0,0	0,0	25,0
№	First, %	Last, %	Top3, %																																						
1	25,0	0,0	25,0																																						
2	0,0	25,0	0,0																																						
6	0,0	25,0	75,0																																						
8	0,0	0,0	75,0																																						
9	25,0	0,0	25,0																																						
5	50,0	25,0	50,0																																						
7	0,0	0,0	25,0																																						
3	0,0	25,0	0,0																																						
4	0,0	0,0	25,0																																						

Nº	First, %	Last, %	Top3, %
1	25,0	0,0	25,0
2	0,0	25,0	0,0
3	0,0	25,0	0,0
4	0,0	0,0	25,0
5	50,0	25,0	50,0
6	0,0	25,0	75,0
7	0,0	0,0	25,0
8	0,0	0,0	75,0
9	25,0	0,0	25,0

Calculated_input.txt

Nº	First, %	Last, %	Top3, %
1	0,0	0,0	25,0
22	25,0	0,0	0,0
2	50,0	0,0	25,0
3	0,0	50,0	25,0
4	0,0	0,0	50,0
5	0,0	0,0	50,0
6	25,0	0,0	0,0
7	0,0	25,0	0,0
8	0,0	0,0	25,0
9	0,0	0,0	50,0
11	0,0	25,0	25,0
19	0,0	0,0	25,0

4 disordered.txt

Nº	First, %	Last, %	Top3, %
1	28,6	0,0	0,0
2	0,0	14,3	0,0
3	14,3	0,0	28,6
4	14,3	28,6	28,6
5	0,0	14,3	28,6
34	14,3	0,0	0,0
6	14,3	0,0	57,1
7	14,3	14,3	42,9
8	0,0	0,0	42,9
23	0,0	0,0	28,6
45	0,0	28,6	0,0
9	0,0	0,0	28,6
43	0,0	0,0	14,3

ordered.txt

Nº	First, %	Last, %	Top3, %
1	28,6	0,0	0,0
2	0,0	14,3	0,0
3	14,3	0,0	28,6
4	14,3	28,6	28,6
5	0,0	14,3	28,6
6	14,3	0,0	57,1
7	14,3	14,3	42,9
8	0,0	0,0	42,9
9	0,0	0,0	28,6
23	0,0	0,0	28,6

34	14,3	0,0	0,0
43	0,0	0,0	14,3
45	0,0	28,6	0,0
Calculated_input.txt			
Nº	First, %	Last, %	Top3, %
20	0,0	0,0	50,0
1	0,0	0,0	25,0
2	0,0	0,0	25,0
3	0,0	0,0	25,0
6	50,0	25,0	0,0
7	25,0	50,0	0,0
8	0,0	0,0	25,0
9	0,0	0,0	25,0
10	0,0	0,0	50,0
13	25,0	0,0	0,0
14	0,0	25,0	0,0
15	0,0	0,0	50,0
19	0,0	0,0	25,0