



Министерство науки и высшего образования Российской
Федерации

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский
университет)» (МГТУ им. Н.Э. Баумана)**

Лабораторная Работа №2 «Записи с вариантами, обработка таблиц» Вариант №2

Студент **Шахнович Дмитрий Сергеевич**

Группа **ИУ7-22Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент Шахнович Д.С.

Оценка _____

2023 г.

Описание условия задачи

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами (объединениями)). Упорядочить данные в ней по возрастанию ключей, двумя алгоритмами сортировки, где ключ – любое невариантное поле (по выбору программиста), используя: а) саму таблицу, б) массив ключей. (Возможность добавления и удаления записей в ручном режиме обязательна).

Ввести список стран, содержащий название страны, столицу, материк, необходимость наличия визы, время полета до страны, минимальную стоимость отдыха, основной вид туризма (1: экскурсионный - количество объектов, их основной вид (природа, история, искусство); 2: пляжный – основной сезон, температура воздуха и воды; 3: спортивный – вид спорта (горные лыжи, серфинг, восхождения, сплав)). Вывести список стран на выбранном материке, со стоимостью ниже указанной, где можно заняться указанным видом спорта.

Техническое задание

Исходные данные:

Файл: В файле содержатся данные о странах, каждое поле записано с новой строки.

В меню действия задаются вводом целых чисел 0-9. Внутри пунктов добавления элемента или ввода названия файла могут вводиться строковые литералы.

Выходные данные:

Программа выдает текущую таблицу, отсортированную по цене путешествия, таблицу стран на выбранном материке с ценой ниже указанной, где можно заняться выбранным спортом, сравнение скоростей работы

сортировки хоара и сортировки пузырьком в случаях обычного массива и массива с дополнительным массивом индексов.

Описание задания:

Чтение, сохранение, вывод, сортировка массива стран, а также поиск по этому массиву.

Способы обращения к программе:

Запуск программы через терминал, затем управление программой с помощью меню. Пункты меню:

1 — Ввести файл для работы с ним. После вызова пункта меню пользователь вводит имя файла. После его спрашивают нужно ли считать данные из файла.

2 — Вывести текущий массив стран.

3 — Добавить в конец массива новую страну. После вызова пункта меню пользователь вводит по одному полю страны по запросу программы.

4 — Удалить из массива страну по номеру. После вызова пункта меню пользователь вводит номер строки.

5 - Вывести список стран на выбранном материке, со стоимостью ниже указанной, где можно заняться указанным видом спорта. После вызова пункта меню пользователь вводит материк, цену и вид спорта.

6 — Сохранить таблицу в файл. Таблица сохраняется в файл введенный в пункте 1.

7 — Отсортировать таблицу в виде массива с дополнительным массивом индексов.

8 — Отсортировать таблицу как обычный массив стран

9 — Вывести времена сортировок пузырьком и хоара в случаях массива с индексами и без него, а также вывести объём памяти, занимаемый массивами.

0 — Выход из программы

Аварийные ситуации:

- 1) В меню введен код, которого нет;
Сообщение: «Некорректный код.»
- 2) Ошибка ввода/вывода;
Сообщение: «Ошибка ввода/вывода.»
- 3) Попытка удалить элемент, которого нет;
Сообщение: «Выход за границы.»
- 4) Введены строковые литералы при запросе числа или число вне указанного диапазона;
Сообщение: «Некорректное значение.»
- 5) Не удалось открыть введенный файл;
Сообщение: «Ошибка с открытием файла.»
- 6) В файле слишком много стран(>10000);
Сообщение: «Слишком большой файл.»
- 7) Ввод пустой строки;
Сообщение: «Пустая строка.»
- 8) Ввод строки превышающей указанный размер;
Сообщение: «Ошибка длины строки.»
- 9) Попытка добавить элемент в полный массив;
Сообщение: «Ошибка длины массива.»

Описание структур данных

```
struct country_t  
{  
    char name[NAME_SIZE + 1];
```

```

char capital[CAPITAL_SIZE + 1];
char land[CAPITAL_SIZE + 1];
int visa;
int ftime;
int price;
vacation_t vacation_type;
union
{
struct excursion_t excursion;
struct beach_t beach;
sport_t sport;
} vacation;
};

```

Структура страны

- Name - Название страны, 15 символов.
- Capital — Название столицы, 15 символов.
- Land — Материк, 18 символов
- Visa — Необходимость наличия визы(1 — нужна, 0 — нет)
- Ftime — Время полета до страны.
- Price — Цена путешествия
- Vacation_type — Тип отдыха(0 — Экскурсии, 1 — Спортивный, 2 -
Пляжный)
- Vacation - Вариантная часть, которая зависит от Vacation_type
 - o Excursion - Состоит из структур

```

struct excursion_t
{
size_t obj_cnt;
obj_t type;
};

```

- o obj_count — Количество объектов

- type — Вид объектов(0 — Природные, 1 — Исторические, 2 — Культурные)
- Beach — Состоит из структуры:

```
struct beach_t
{
season_t season;
double water_temp;
double air_temp;
};
```

- season — Лучший сезон для отдыха(0 — Зима, 1 — Весна, 2 — Лето, 3 — Осень)
- water_temp — Температура воды
- air_temp — Температура воздуха
 - Sport — Вид спорта(0 — Лыжи, 1 — Сёрфинг, 2 — Восхождение, 3 — Сплав).

```
struct country_arr_t
{
size_t size;
struct country_t arr[MAX_ARRAY_SIZE];
};
```

Структура массива стран

- Size - Количество элементов массива.
- Arr - Массив стран, размер — 10000 элементов.

```
struct country_ind_arr_t
{
size_t size;
struct country_t arr[MAX_ARRAY_SIZE];
size_t indexes[MAX_ARRAY_SIZE];
};
```

```
};
```

Структура массива стран с дополнительным массивом индексов

- Size - Количество элементов массива.
- Arr - Массив стран, размер — 10000 элементов.
- Indexes — Массив индексов стран — 10000 элементов.

```
char fname[MAX_ARRAY_SIZE];
```

Название файла в main, 32 символа

Описание алгоритма

1. Вывести пользователю меню и ожидать ввода номера команды;
2. В случае команды добавления или удаления элемента из массива, проверить его размеры и провести операцию;
3. В случае одной сортировок отсортировать массив;
4. В случае выбора замеров, создать копии массивов, отсортировать их, замерив каждое время сортировки;
5. В случае открытия файла или записи в него, проверить возможность открытия файла в нужном режиме и провести открытие/запись;
6. В случае выхода закрыть файл и завершить программу.

Тестовые данные

Позитивные тесты			
№	Описание	Вход	Выход
1	Удалить в массиве из одного элемента	4 1	Пустой массив
2	Удалить первый элемент	4 1	Массив из $n - 1$ сдвинувшихся влево элементов
3	Удалить последний элемент	4 n	Массив из $n - 1$ элементов
4	Добавить элемент в пустой массив	3 {Корректная страна}	Массив из введённого элемента
5	Добавить в массив из 9999 элементов	3 {Корректная страна}	Массив из 10000 стран
6	Ввести файл с названием в 32 символа	1 {Название из 32 символов}	Файл успешно считан
7	Вывести пустой массив	2	Пустая таблица
8	Вывести не пустой массив	2	N стран
9	Провести поиск в массиве, где нет искомых стран	5 {корректные данные поиска}	Пустая таблица
10	Сохранить таблицу в файл	6	Таблица успешно сохранена
11	Отсортировать	7	Отсортированная таблица

	отсортированный массив		
12	Отсортировать отсортированный в обратном порядке массив	7	Отсортированная таблица
13	Отсортировать случайный массив	7	Отсортированная таблица
14	Отсортировать пустой массив	7	Пустая таблица
15	Отсортировать массив из одного элемента	7	Один элемент
16	Вывести таблицу эффективности сортировок для случайного массива	9	Таблица с временами сортировок
Негативные тесты			
1	Ввести пустую строку		Пустая строка.
2	Ввести неверный код в меню	12	Некорректный ввод.
3	При запросе числа ввести литерал	4 1a	Некорректное значение.
4	Ввести слишком длинную строку	1 {33 символа «а»}	Ошибка длины строки..
5	Ввести несуществующее название файла	1 {Не суц. файл}	Ошибка с открытием файла.
6	Ввести название файла, с слишком большим числом стран	1 {Файл с кол-вом стран > 10000}	Ошибка с открытием файла.
7	Попытаться удалить	4	Выход за границы.

	несуществующий элемент	10001	
8	Попытаться добавить в полный массив	3	Максимальный размер таблицы(10000 стран)

Замеры сортировок

Замеры сортировок проводились следующим образом: с помощью скрипта создавался файл со n-м количеством случайных стран, затем в программе замера это файл считывается в массив. Для каждой сортировки в 100 итерациях считанный массив копируется во временный, который после сортируется, измеряя время только алгоритма сортировки. Время всех итераций суммируется и делится на количество сортировок(в данном случае 100), таким образом считается среднее арифметическое время сортировки.

Кол-во элементов	Массив без индексов		Массив с индексами	
	Быстрая сортировка,мкс	Пузырёк,мкс	Быстрая сортировка,мкс	Пузырёк,мкс
100	4	40	2	17
500	31	985	18	482
1000	74	3535	41	2070
2000	168	13140	108	8566
4000	496	50585	343	31004
7500	1067	175807	953	117238

Кол-во элементов	Массив без индексов,байт	Массив с индексами,байт
100	9608	10408
500	48008	52008
1000	96008	104008
2000	192008	208008
4000	384008	416008
7500	720008	780008

Отношение времени сортировки в массиве с индексами к времени в обычно массиве

Кол-во элементов	Быстрая сортировка	Пузырёк
100	2.0	2.4
500	1.7	2.1
1000	1.8	1.7
2000	1.6	1.5
4000	1.5	1.6
7500	1.2	1.5

Как видно использование массива индексов ускоряет алгоритм сортировки, особенно в тех алгоритмах, где происходит частый обмен значений, как в пузырьковой сортировке, где использование таблицы ключей дает выигрыш по времени в 1.5-1.7 раз. При этом количество затрачиваемой памяти увеличивается всего на 8%, что во многих задачах может быть несущественным, поэтому рационально использовать таблицу ключей с типами большого размера.

Ответы на вопросы

1. *Как выделяется память под вариантную часть записи?*

Память под вариантную часть выделяется по наибольшему элементу, входящему в эту часть.

2. *Что будет, если в вариантную часть ввести данные, несоответствующие описанным?*

Программа аварийно завершит работу.

3. *Кто должен следить за правильностью выполнения операций с вариантной частью записи?*

Программист должен следить за тем, чтобы некорректные данные не доходили вариантной записи, например заканчивая ввод пользователя сообщением об ошибках.

4. *Что представляет собой таблица ключей, зачем она нужна?*

Таблица ключей — дополнительный массив, в котором содержатся индексы элементов массива элементов. Может использоваться для ускорения операций, требующих перестановок элементов массива, так как переставить местами два числа часто бывает быстрее, чем сами элементы.

5. *В каких случаях эффективнее обрабатывать данные в самой таблице, а когда – использовать таблицу ключей?*

При небольших элементах эффективнее обрабатывать сами данные. Использовать таблицу ключей рационально использовать тогда когда размер элемента >> размер индекса.

6. *Какие способы сортировки предпочтительнее для обработки таблиц и почему?*

Лучше всего использовать те сортировки в которых операций сравнений больше чем операций перестановок, так как такие операции чаще всего работают быстрее на больших типах данных.

Выводы

Периодически перед программистом встают задачи, когда ему рационально использовать записи с вариантной частью. В таком случае программисту важно следить за правильностью этой самой вариантной частью. Также часто приходится работать с большими данными и поэтому важно уметь ускорить обработку таких данных, например используя таблицу ключей или более быстрые методы обработки, например использование алгоритма хоара для сортировки.