

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна

Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Індивідуальне завдання № 1
з курсу «Алгоритми і структури даних»
на тему: «Односпрямовані списки»

Виконала: студентка 2 курсу групи МФ-22

напряму підготовки (спеціальності)

Комп'ютерні науки

122 Комп'ютерні науки

Мухачова В.В.

Завдання:

Спортивна олімпіада. Є набір текстових файлів. У першому міститься інформація про країни-учасниці спортивної олімпіади у вигляді набору рядків, у кожному з яких – назва країни та перелік видів спорту, в яких вона бере участь. Інші файли (їх стільки, скільки видів спорту) – це результати змагань: кожен називається назвою виду спорту і містить перелік країн у порядку зайнятих ними місць у цьому виді спорту. Потрібно створити 1) список видів спорту із зазначенням загальної кількості команд-учасниць; 2) список списків: список країн в алфавітному порядку та для кожної країни список видів спорту та зайнятого місця у цьому виді спорту, впорядкований за зростанням місць. Забезпечити операції додавання, видалення, редагування та розумного пошуку інформації в обох списках, а також виведення інформації в розумному вигляді (наприклад, країна, вид спорту та зайняте місце із загальної кількості команд у цьому виді спорту).

Аналіз коду:

```
typedef struct sport
{
    char name[50];
    int num_teams;
    struct sport *next;
}sport;
```

Структура sport зберігає інформацію про спорт: ім'я, кількість команд та вказівник на наступний елемент.

```
typedef struct country_sl
{
    char sport_name[50];
    int place;
    struct country_sl *next_sport;
}country_sl;
```

Структура country_sl зберігає інформацію про спорт: назву спорту, місце та вказівник на наступний елемент.

```
typedef struct country
{
    char name[50];
    country_sl *sports_list;
    struct country *next;
}country;
```

Структура `country` зберігає інформацію про країну: назву, список видів спорту та вказівник на наступну країну.

Опис функцій програми:

1. Функція `main()` ініціалізує масив для імен видів спорту та змінні, імпортує дані про країни і види спорту, сортує види спорту за місцем, виконує меню для вибору дій користувача, а також звільняє пам'ять наприкінці.
2. Функція `Menu()` виводить список доступних дій для користувача та запитує вибір користувача.
3. Функція `Work()` обробляє вибір користувача (параметр `i`), виконуючи відповідні дії: виведення списків, пошук, редагування, додавання чи видалення країн і видів спорту.
4. Функція `import_countries_from_file()` завантажує дані країн з файлу `countries.txt`, зчитує назву країни та спорт, у якому країна бере участь. Якщо спорт новий, він додається до масиву та до списку спортів країни. Потім країна додається до списку країн, впорядкованого за алфавітом.
5. Функція `add_sport_to_country()` додає спорт до списку спортів країни, вставляючи новий елемент на правильне місце в порядку місця.
6. Функція `import_sports_from_file()` завантажує дані про команди, які беруть участь у кожному виді спорту, з відповідних файлів (наприклад, `спорт.txt`). Вона створює нові елементи в списку видів спорту, зчитує країни та кількість команд, оновлює місце в списку спорту країни та додає інформацію про вид спорту до основного списку.
7. Функція `sort_sports_list_by_place()` сортує список спортів кожної країни за місцем, яке країна зайняла, використовуючи метод сортування вибором. Вона знаходить спорт з найвищим місцем, переставляє його на початок списку, і так повторює для всіх елементів.
8. Функція `search_country()` шукає країну за назвою в списку країн. Якщо країна знайдена, повертається вказівник на неї; якщо ні — виводиться повідомлення про помилку і повертається `NULL`.
9. Функція `search_sport_sl()` шукає спорт у списку спортів країни за назвою. Якщо спорт знайдений, повертається вказівник на відповідний елемент; якщо ні — повертається `NULL`.
10. Функція `search_sport()` шукає спорт у основному списку спортів за назвою. Якщо спорт знайдений, повертається вказівник на відповідний елемент; якщо ні — повертається `NULL`.

11. Функція `Print_sport()` виводить основний список спортів, показуючи ім'я спорту та кількість команд у кожному виді спорту. Якщо список порожній, виводиться повідомлення про помилку.
12. Функція `Print_countries()` виводить список країн. Для кожної країни виводиться її назва, а також викликається функція `Print_sl()`, щоб вивести список спортів, у яких вона бере участь. Якщо список країн порожній, виводиться повідомлення про помилку.
13. Функція `Print_sl()` виводить список спортів для однієї країни. Для кожного спорту виводиться його назва та місце, яке країна зайняла. Якщо список спорів порожній, виводиться повідомлення про помилку.
14. Функція `Search()` дозволяє користувачеві здійснювати пошук за спортом або за країною.
15. Функція `Edit_info_countries()` дозволяє редагувати інформацію про країни та їх участь у різних видах спорту. Вона дає можливість або змінити місце країни в спорті, або додати новий спорт до списку країни. Опис роботи функції:
 - Пошук країни: Спочатку користувач вводить назву країни, яку він хоче редагувати. Якщо країна не знайдена в списку, виводиться повідомлення про помилку.
 - Користувач може вибрати одну з двох операцій:
 - Змінити місце країни в спорті: Виводиться список спортів країни. Користувач вибирає спорт, для якого хоче змінити місце. Місце, яке країна зайняла в цьому спорті, змінюється за допомогою введення нового значення.
 - Додати новий спорт: Користувач може вибрати спорт з наявного списку. Спорт додається до списку спортів країни, і країна також додається до відповідного файлу спорту. Оновлюється кількість команд у цьому спорті в основному списку спортів.
 - Оновлення даних: Після зміни або додавання спорту, відповідні файли (країн і спорту) оновлюються, щоб відобразити нові дані. У файлі країн також зберігається інформація про додані спорти.
16. Функція `Add_country()` додає нову країну до списку. Вона виконує наступні дії:
 - Додавання нової країни: Користувач вводить назву нової країни. Країна додається до списку в алфавітному порядку.

- Додавання спортивів: Після того, як країна була додана, користувач може додавати до неї різні види спорту. Для кожного доданого виду спорту, країна записується в файл цього спорту, а кількість команд у спорті збільшується.
- Оновлення файлів: Після того, як всі зміни внесено, оновлюються файли країн для збереження нових даних.

17. Функція `Add_sports()` додає новий спорт до списку доступних видів спорту та оновлює відповідні дані про країни, що беруть участь у цьому спорті. Опис роботи функції:

- Перевірка наявності спорту: Спочатку перевіряється, чи існує вже спорт з введеною назвою в масиві наявних видів спорту. Якщо такий спорт вже є, виводиться повідомлення про це, і функція припиняє виконання.
- Додавання нового спорту: Якщо спорт не існує, створюється новий об'єкт типу `sport`, заповнюється його назвою та кількістю команд (яка початково дорівнює 1), і він додається до списку спортивів. Також додається назва спорту до масиву `arg_sports`.
- Додавання країни, яка бере участь у спорті: Користувач вводить назву країни, яка бере участь у новому спорті. Перевіряється, чи є така країна в списку. Якщо країна знайдена, до її списку додається новий спорт, і відповідні дані оновлюються в файлі країн. Якщо країна не знайдена, виводиться повідомлення про помилку, і користувач може спробувати ще раз.
- Створення нового файлу для спорту: Створюється файл з назвою, що відповідає спорту (додається розширення `.txt`). У файл записується назва країни, яка бере участь у спорті.

18. Функція `Remove_info_countries()` призначена для видалення країни з усіма її спортивними даними з системи. Опис роботи функції:

- Пошук країни. Визначається, чи є країна у списку.
- Видалення спортивних даних. Для країни видаляються всі спортивні результати, і зменшується кількість команд для кожного спорту.
- Видалення країни зі списку. Країна видаляється зі зв'язного списку країн.
- Оновлення файлів. Видаляється країна з файлів країн та файлів кожного спорту.

19. Функція `Remove_info_sports()` призначена для видалення виду спорту з системи, що включає видалення з усіх країн, списку спортивів, файлів та масивів. Опис роботи функції:

- Перевірка наявності спорту в списку: Спочатку функція перевіряє, чи є спорт у списку. Якщо список порожній, виводиться повідомлення про це, і функція завершується. Користувач вводить назву виду спорту, який хоче видалити.

- Видалення спорту з масиву: Функція проходить по масиву `arr_sports[]`, і якщо знаходить введену назву виду спорту, то видаляє його. Якщо спорт не знаходиться в кінці масиву, то копіюється останній елемент на місце цього спорту, і довжина масиву зменшується.
 - Видалення спорту з основного списку спортів: Функція проходить по списку спортів і шукає спорт для видалення. Якщо спорт знаходиться на першій позиції, список спорту змінюється, і пам'ять, зайнята спортом, звільняється. Якщо спорт знаходиться в середині списку, попередній елемент вказує на наступний, і пам'ять для цього спорту звільняється.
 - Видалення спорту зі списків країн: Функція проходить по кожній країні в списку країн і шукає відповідний спорт у списку спортів цієї країни. Якщо спорт знайдений, він видаляється з цього списку, і пам'ять звільняється.
 - Видалення файлу спорту: Для виду спорту видаляється відповідний текстовий файл, що містить дані про цей спорт, за допомогою `remove()`.
 - Оновлення файлу країн `countries.txt`: Оскільки спорт видаляється, потрібно оновити файл країн. Функція створює тимчасовий файл (`temp.txt`), у який переписуються всі країни з їхніми видами спорту, за винятком видаленого спорту. Всі країни, що містять видалений спорт у своєму списку, обробляються так, щоб він більше не фігурував у списку. Після цього старий файл `countries.txt` видаляється, а тимчасовий файл перейменовується в `countries.txt`.
20. Функція `Update_file()` видаляє всі рядки з файлу, що містять назву країни. Вона читає вхідний файл, перевіряє, чи є згадка країни, і переписує рядки без згадок у новий тимчасовий файл. Після цього старий файл видаляється, а тимчасовий файл перейменовується в оригінальний.
21. Функція `append_to_sport_file()` додає нову країну до відповідного файлу спорту. Вона створює ім'я файлу за допомогою назви спорту, відкриває файл в режимі додавання, і записує назву країни в новому рядку.
22. Функція `update_countries_file()` додає новий спорт для існуючої країни або додає нову країну з її першим видом спорту до файлу `countries.txt`.
23. Функція `gets_temp()` зчитує введену строку з консолі, перевіряючи її на валідність за кількома умовами:
- Перевірка на порожній ввід: Якщо користувач натискає Enter без вводу, виводиться повідомлення і запит повторюється.
 - Перевірка на некоректні символи: Введена строка має містити тільки літери (великі та малі), пробіли, апострофи (') і тире (-). Якщо введений символ не відповідає жодному з цих критеріїв, виводиться помилка.

- Обрізання зайвих символів: Наприкінці введеного рядка обривається символ нового рядка (\n), щоб не виникло проблем із подальшою обробкою.
- 24. Функція `get_number_in_range()` зчитує число з консолі та перевіряє, чи воно знаходиться в заданому діапазоні. Якщо введення не є числом або виходить за межі діапазону, запит повторюється.
- 25. Функція `open_check()` перевіряє, чи вдалося успішно відкрити файл. Якщо файл не відкривається, виводиться повідомлення про помилку.
- 26. Функція `close_check()` перевіряє, чи вдалося успішно закрити файл. Якщо при закритті файлу виникає помилка, виводиться повідомлення про помилку.
- 27. Функція `allocate_memory()` виділяє пам'ять за допомогою `malloc()` і перевіряє, чи вдалося це зробити.
- 28. Функція `free_sport_list()` призначена для звільнення пам'яті, яка була виділена для списку спортивних об'єктів.
- 29. Функція `free_country_list()` звільняє пам'ять, виділену для списку країн, а також для списку спортів кожної країни.

Приклад роботи програми:

Приклад файлів з вхідними даними

```
main > cat output > countries.txt
Ukraine: swimming, basketball, athletics, volleyball, tennis, dancing, skating
Korea: swimming, fencing, gymnastics, tennis, judo
Norway: athletics, swimming, volleyball, cycling
China: fencing, gymnastics, athletics, judo, cycling
Brazil: swimming, athletics, basketball, cycling, volleyball
Japan: judo, tennis, volleyball
France: fencing, athletics, tennis, cycling, swimming
Australia: dancing
Peru: tennis, swimming
Laos: skating, dancing
Canada: skating, athletics
Germany: basketball
United Kingdom: racing, cycling, judo, football
```

Файл countries.txt

```
AiSD > indiv > output > ≡ skating.txt
1   Ukraine
2   Laos
3   Canada
```

Файл skating.txt

При запуску програми користувач отримує меню:

```
Welcome to the menu!
What would you like to do?
1. Output the list of sports.
2. Output the list of countries and sports they participate in.
3. Output both the list of sports and the list of countries.
4. Search the information.
5. Edit the information of a country
6. Add a new country
7. Add a new sport
8. Remove a country
9. Remove a sport
```

Меню програми

Користувач може вибрати між трьома варіантами виводу, наприклад вивести і список видів спорту і список країн зі списками видів спорту в яких вони приймають участь:

```
Sport      || Number of teams
football   || 1
racing     || 1
cycling    || 5
judo       || 4
gymnastics || 2
fencing    || 3
skating    || 3
dancing    || 3
tennis     || 5
volleyball || 4
athletics  || 6
basketball || 3
swimming   || 6
Country: |Australia|
Name of the sport || Place taken
dancing          || 2
-----
Country: |Brazil|
Name of the sport || Place taken
swimming          || 1
basketball        || 1
athletics         || 2
cycling           || 2
volleyball        || 2
-----
Country: |Canada|
Name of the sport || Place taken
skating           || 3
athletics         || 6
```

Приклад виведення даних

Користувач може відфільтрувати інформацію за країною або видом спорту:

```
54 remove a sport
4
What would you like to find?
1. Search by sport
2. Search by country.
2
Please, input the name of the country
Ukraine
Country: |Ukraine|
Name of the sport || Place taken
dancing           || 1
skating           || 1
swimming          || 2
basketball        || 2
athletics         || 3
tennis            || 3
volleyball        || 4
-----
```

Приклад роботи функції пошуку

Користувач також може додати нову країну до списків:

```
6
Enter the name of the new country:
Thailand
Do you want to add a sport to 'Thailand'? (1 for Yes, 0 for No):
1
Available sports:
1. swimming
2. basketball
3. athletics
4. volleyball
5. tennis
6. dancing
7. skating
8. fencing
9. gymnastics
10. judo
11. cycling
12. racing
13. football
Enter the number of the sport:
6
```

```

4
What would you like to find?
1. Search by sport
2. Search by country.
2
Please, input the name of the country
Thailand
Country: |Thailand|
Name of the sport || Place taken
dancing          || 1
-----

```

Приклад додавання нової країни.

Користувач може додати новий спорт:

```

7. Add a new sport
8. Remove a country
9. Remove a sport
7
Enter the name of the new sport: running
Enter the name of a country that participates in this new sport: Ukraine
Country 'Ukraine' will participate in the new sport.
File 'running.txt' created successfully.
Would you like to continue? Enter 'y' if yes, 'n' if no

```

```

AiSD > indiv1 > output > ≡ running.txt
1   Ukraine
2

```

Файл running.txt після створення

```

AiSD > indiv1 > output > ≡ countries.txt
1   Ukraine: swimming, basketball, athletics, volleyball, tennis, dancing, skating, running
2   Korea: swimming, fencing, gymnastics, tennis, judo
3   China: fencing, gymnastics, athletics, judo
4   Japan: judo, tennis, volleyball
5   Australia: dancing

```

Файл countries.txt після додавання нового спорту

Користувач може видалити країну:

```
8. Remove a country
9. Remove a sport
8
Which country would you like to remove?
Brazil
Would you like to continue? Enter 'y' if yes, 'n' if no
y
```

```
What would you like to find?
1. Search by sport
2. Search by country.
2
Please, input the name of the country
Brazil
Country not found.
Would you like to continue? Enter 'y' if yes, 'n' if no

```

Приклад видалення країни

Також користувач може видалити спорт:

```
9. Remove a sport
9
Which sport would you like to remove?
tennis
Would you like to continue? Enter 'y' if yes, 'n' if no
y
```

```
What would you like to find?
1. Search by sport
2. Search by country.
1
Please, input the name of the sport
tennis
Sorry, sport not found.
Would you like to continue? Enter 'y' if yes, 'n' if no

```

Приклад видалення спорту

Оригинальный код програми:

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
#define temp_length 100

typedef struct sport
{
    char name[50];
    int num_teams;
    struct sport *next;
}sport;

typedef struct country_sl
{
    char sport_name[50];
    int place;
    struct country_sl *next_sport;
}country_sl;

typedef struct country
{
    char name[50];
    country_sl *sports_list;
    struct country *next;
}country;

int Menu();
void Work(int i, country **country_head, sport **sport_head, char arr_sports[][temp_length], int
arr_length);

void import_countries_from_file(country **countries_head, char arr_sports[][temp_length], int*
arr_length);
void add_sport_to_country(country *c, const char *sport_name, int place);
void import_sports_from_file(sport **sports_head, char arr_sports[][temp_length], int arr_length, country
**countries_head);
void sort_sports_list_by_place(country **countries_head);
country* search_country(country *head, const char *country_name);
country_sl* search_sport_sl(country_sl *head, const char *sport_name);
sport* search_sport(sport *head, const char *sport_name);

void Print_sport(sport *head);
void Print_countries(country *head);
void Print_sl(country_sl *head);

void Search(country *country_head, sport *sport_head);
void Edit_info_countries(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length);
```

```

void Add_country(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length);
void Add_sports(sport **sport_head, country **country_head, char arr_sports[][temp_length], int
*arr_length);
void Remove_info_countries(country **country_head, sport **sport_head, char arr_sports[][temp_length],
int arr_length);
void Remove_info_sports(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length);

void Update_file(const char *filename, const char *country_to_remove);
void append_to_sport_file(const char *sport_name, const char *country_name);
void update_countries_file(const char *country_name, const char *new_sport);

void gets_temp(char *temp);
FILE *open_check(FILE *fp, char str[30], char ch[3]);
void close_check(FILE *fp);
int get_number_in_range (int left, int right);
void* allocate_memory(size_t size);
void free_sport_list(sport *sports_head);
void free_country_list(country *countries_head);

int main()
{
    char sports_arr[50][temp_length] = {};
    int sports_arr_length = 0;

    country *countries_head = NULL;
    sport *sports_head = NULL;

    import_countries_from_file(&countries_head, sports_arr, &sports_arr_length);
    import_sports_from_file(&sports_head, sports_arr, sports_arr_length, &countries_head);
    sort_sports_list_by_place(&countries_head);

    int menu_switch;
    char ch = 'y';
    while (ch=='y' || ch=='Y')
    {
        menu_switch = Menu();
        Work(menu_switch, &countries_head, &sports_head, sports_arr, sports_arr_length);
        printf("Would you like to continue? Enter 'y' if yes, 'n' if no \n");
        ch = getchar();
        getchar();
    }

    free_country_list(countries_head);
    free_sport_list(sports_head);
    return 0;
}

```

```

int Menu()
{
    int i = 0;
    printf("Welcome to the menu!\n");
    printf("What would you like to do?\n");
    printf("1. Output the list of sports. \n2. Output the list of countries and sports they participate in.\n");
    printf("3. Output both the list of sports and the list of countries.\n");
    printf("4. Search the information.\n");
    printf("5. Edit the information of a country\n");
    printf("6. Add a new country\n");
    printf("7. Add a new sport\n");
    printf("8. Remove a country\n");
    printf("9. Remove a sport\n");

    i = get_number_in_range (1, 9);
    return i;
}

```

```

void Work(int i, country **country_head, sport **sport_head, char arr_sports[][temp_length], int
arr_length)
{
    switch (i)
    {
        case 1:
            Print_sport(*sport_head);
            break;
        case 2:
            Print_countries(*country_head);
            break;
        case 3:
            Print_sport(*sport_head);
            Print_countries(*country_head);
            break;
        case 4:
            {
                Search(*country_head, *sport_head);
                break;
            }
        case 5:
            {
                Edit_info_countries(country_head, sport_head, arr_sports, &arr_length);
                break;
            }
        case 6:
            {
                Add_country(country_head, sport_head, arr_sports, &arr_length);
                break;
            }
        case 7:

```

```

    {
        Add_sports(sport_head, country_head, arr_sports, &arr_length);
        break;
    }
    case 8:
    {
        Remove_info_countries(country_head, sport_head, arr_sports, arr_length);
        break;
    }
    case 9:
    {
        Remove_info_sports(country_head, sport_head, arr_sports, &arr_length);
        break;
    }
    default:
        printf("Sorry, that option is not found\n");
}
}
}

```

//завантажує дані країн з файлу "countries.txt"

```
void import_countries_from_file(country **countries_head, char arr_sports[][temp_length], int* arr_length)
```

```

{
    FILE *infile = NULL;
    infile = open_check(infile, "countries.txt", "r");
    char *temp = (char*)allocate_memory(temp_length*sizeof(char));

    while (fgets(temp, temp_length, infile) != NULL)
    {
        //відкриває файл і читає назву країни на початку кожної строки
        country *temp_country = (country *)allocate_memory(sizeof(country));
        temp_country->sports_list = NULL;

        char *token = strtok(temp, ":");
        if(token!=NULL)
        {
            token[strcspn(token, "\n")] = '\0';
            strcpy(temp_country->name, token);
            token = strtok(NULL, " :");

            //читає назву кожного спорту в якому країна приймає участь
            while (token != NULL) {
                token[strcspn(token, "\n")] = '\0';

                //перевіряє чи спорт вже існує в масиві з назвами спортив
                int sport_exists = 0;
                for (int i = 0; i < *arr_length; i++) {
                    if (strcmp(arr_sports[i], token) == 0) {

```

```

        sport_exists = 1;
        break;
    }
}

//якщо не існує, додає в масив і додає до листу sport's list країни
if (!sport_exists) {
    strcpy(arr_sports[*arr_length], token);
    (*arr_length)++;
    add_sport_to_country(temp_country, token, *arr_length);
}
else {
    //якщо вже існує, то просто додає до листу sport's list країни
    add_sport_to_country(temp_country, token, 0);
}

token = strtok(NULL, " :");
}
}

//додає країну з врахуванням впорядкування за алфавітом
if (*countries_head == NULL || strcmp(temp_country->name, (*countries_head)->name) < 0)
{
    temp_country->next = *countries_head;
    *countries_head = temp_country;
}
else
{
    country *current = *countries_head;
    while (current->next != NULL && strcmp(current->next->name, temp_country->name) < 0)
    {
        current = current->next;
    }
    temp_country->next = current->next;
    current->next = temp_country;
}
}

free(temp);
close_check(infile);
}

//додає спорт до листу спортів країни
void add_sport_to_country(country *c, const char *sport_name, int place)
{
    //створює нову ноду
    country_sl *new_sport = (country_sl *)allocate_memory(sizeof(country_sl));
    strcpy(new_sport->sport_name, sport_name);
    new_sport->place = place;
}

```



```

//вставляє на правильне місце в списку
if (c->sports_list == NULL || c->sports_list->place <= place)
{
    new_sport->next_sport = c->sports_list;
    c->sports_list = new_sport;
}
else
{
    country_sl *current = c->sports_list;
    while (current->next_sport != NULL && current->next_sport->place > place)
    {
        current = current->next_sport;
    }
    new_sport->next_sport = current->next_sport;
    current->next_sport = new_sport;
}
}

//завантажує дані про команди які приймають участь в кожному спорті
void import_sports_from_file(sport **sports_head, char arr_sports[][temp_length], int arr_length, country
**countries_head)
{
    FILE *infile = NULL;

    int buf = 0;
    char *temp = (char *)allocate_memory(temp_length * sizeof(char));

    //проходить по кожному спорту в масиві
    for(int i = 0; i < arr_length; i++)
    {
        //створює нову ноду в основному листі спортів
        sport *temp_sport = (sport*)allocate_memory(sizeof(sport));

        buf = 0;
        strcpy(temp, arr_sports[i]);

        //Відкриває файл з назвою "спорт.txt"
        infile = open_check(infile, strcat(temp, ".txt"), "r");

        strcpy(temp_sport->name, *(arr_sports+i));

        //читає кожну строку з назвами країн
        while (fgets(temp, temp_length, infile) != NULL)
        {
            //зберігає кількість команд
            buf++;

            country *temp_country = (country *)allocate_memory(sizeof(country));

```

```

    char *token;
    token = strtok(temp, "\n");
    if(token!=NULL)
    {
        token[strcspn(token, "\n")] = '\0';
    }
    //знаходить країну з такою назвою, знаходить спорт в листі спортів країни і змінює місце
    temp_country = search_country(*countries_head, token);
    country_sl *new_sport = (country_sl *)allocate_memory(sizeof(country_sl));
    new_sport = search_sport_sl(temp_country->sports_list, arr_sports[i]);
    new_sport->place = buf;
}

//додає нову ноду на початок списку
temp_sport->num_teams = buf;
temp_sport->next = *sports_head;
*sports_head = temp_sport;
close_check(infile);
}
free(temp);
}

//сортує лист спортів країни за місцем яке країна заняла
//використовується сортування вибором
void sort_sports_list_by_place(country **countries_head)
{
    country *current_country = *countries_head;
    //проходить по кожній країні
    while(current_country!=NULL)
    {
        country_sl *sorted_head = NULL;
        country_sl *unsorted_head = current_country->sports_list;

        while (unsorted_head != NULL) {
            country_sl *min_sport = unsorted_head;
            country_sl *prev_min = NULL;
            country_sl *prev = NULL;

            //знаходить спорт з найвищим місцем
            country_sl *current = unsorted_head;
            while(current!=NULL)
            {
                if (current->place > min_sport->place) {
                    min_sport = current;
                    prev_min = prev;
                }
                prev = current;
                current = current->next_sport;
            }
        }
    }

```

```

    }

    //реорганізує лист
    if (prev_min)
    {
        prev_min->next_sport = min_sport->next_sport;
    }
    else
    {
        unsorted_head = min_sport->next_sport;
    }

    min_sport->next_sport = sorted_head;
    sorted_head = min_sport;
}
//змінює вказівник на відсортований лист
current_country->sports_list = sorted_head;
current_country = current_country->next;
}
}

//шукає країну за назвою
country* search_country(country *head, const char *country_name)
{
    if(head==NULL)
        printf("error of search country/n");
    while (head != NULL)
    {
        if(strcmp(head->name, country_name)==0)
            return head;
        else
            head = head->next;
    }
    printf("Country not found.\n");
    return NULL;
}

//шукає спорт в листі спортів країни за назвою
country_sl* search_sport_sl(country_sl *head, const char *sport_name)
{
    if(head==NULL)
        printf("error of search sport/n");
    while (head != NULL)
    {
        if(strcmp(head->sport_name, sport_name)==0)
            return head;
        else
            head = head->next_sport;
    }
}

```

```

    return NULL;
}

//шукає спорт в основному листі спортів
sport* search_sport(sport *head, const char *sport_name)
{
    if(head==NULL)
        printf("erorr of search sport/n");
    while (head != NULL)
    {
        if(strcmp(head->name, sport_name)==0)
            return head;
        else
            head = head->next;
    }

    return NULL;
}

//виводить основний список спортів
void Print_sport(sport *head)
{
    if(head==NULL)
        printf("erorr of print/n");

    printf("Sport    || Number of teams\n");
    while (head != NULL)
    {
        printf("%-11s || %d\n", head->name, head->num_teams);
        head = head->next;
    }
}

//виводить список країн
void Print_countries(country *head)
{
    if(head==NULL)
        printf("erorr of print/n");
    while (head != NULL)
    {
        printf("Country: |%s|\n", head->name);
        Print_sl(head->sports_list);
        head = head->next;
    }
}

//виводить список спортів країни
void Print_sl(country_sl *head)

```

```

{
    if(head==NULL)
    {
        printf("error of print\n");
        return;
    }
    printf("Name of the sport || Place taken\n");
    while (head != NULL)
    {
        printf("%-17s || %d\n", head->sport_name, head->place);
        head = head->next_sport;
    }
    printf("-----\n");
}

```

//пошук інформації

```
void Search(country *country_head, sport *sport_head)
```

```

{
    int search_count = 0;
    country *temp_country = (country *)allocate_memory(sizeof(country));
    sport *temp_sport = (sport*)allocate_memory(sizeof(sport));
    char *temp = (char*)allocate_memory(temp_length*sizeof(char));
    printf("What would you like to find?\n1. Search by sport\n2. Search by country.\n");
    search_count = get_number_in_range (1, 2);
    int buf = 0;

```

//дає вибір - пошук за спортом або за країною

```
switch (search_count)
```

```

{
    case 1:
        printf("Please, input the name of the sport\n");
        gets_temp(temp);
        printf("Sport: |%s|\n", temp);

        temp_sport = search_sport(sport_head, temp);
        if(temp_sport==NULL)
            return;
        buf = temp_sport->num_teams;
        //виводить список країн які приймають участь і місце яке вони зайняли
        if(buf == 0)
            printf("Sorry, the sport isn't found or no countries are participating.\n");
        else
        {
            printf("List of %d countries participating: ", buf);
            temp_country = country_head;
            while(temp_country!=NULL)
            {
                if(search_sport_sl(temp_country->sports_list, temp)!=NULL)

```

```

        printf("%s | ", temp_country->name);
        temp_country = temp_country->next;
    }
    printf("\n");
}
break;

//виводить країну та список спортив в яких вона приймала участь
case 2:
    printf("Please, input the name of the country\n");
    gets_temp(temp);
    temp_country = search_country(country_head, temp);
    if(temp_country==NULL)
        return;
    printf("Country: |%s|\n", temp_country->name);
    Print_sl(temp_country->sports_list);

default:

    break;
}

free(temp);
}

//редагує інформацію про країну
//дає можливість або змінити місце країни в спорті де вона приймає участь
//або додати новий спорт
void Edit_info_countries(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length)
{
    country *curr_country;
    country_sl *prev_sl;
    char *temp = (char*)allocate_memory(temp_length*sizeof(char));

    int buf = 0, count = 0;

    if (*country_head == NULL)
    {
        printf("The country list is empty.\n");
        return;
    }

    printf("Which country would you like to edit?\n");
    gets_temp(temp);
    curr_country = search_country(*country_head, temp);

    printf("What would you like to do now?\n");

```

```

printf("1. Change a place that a country took in a sport\n");
printf("2. Add a new sport to the list\n");
buf = get_number_in_range (1,2);

//знаходить спорт змінює місце
if(buf==1)
{
    Print_sl(curr_country->sports_list);
    printf("Which sport would you like to change?\n");
    gets_temp(temp);
    prev_sl = curr_country->sports_list;

    while (prev_sl!=NULL)
    {
        if(strcmp(prev_sl->sport_name, temp)==0)
        {
            printf("The country took %d place.\n", prev_sl->place);
            printf("Input the place.\n");
            count = get_number_in_range (1, __INT_MAX__);
            prev_sl->place = count;
            break;
        }
        prev_sl = prev_sl->next_sport;
    }
}

//показує список спортів в яких вже приймає участь
//i показує список всіх спортів
if(buf==2)
{
    Print_sl(curr_country->sports_list);
    printf("The list of sports now:\n");
    for(int i = 0; i < *arr_length; i++)
        printf("%d. %s\n",i+1, arr_sports[i]);

    printf("Input the number of the sport.\n");
    count = get_number_in_range (1, *arr_length) - 1;

    printf("Which place does this country get in this sport?\n");
    buf = get_number_in_range (0, __INT_MAX__);

    //додає спорт до списку спортів країни
    add_sport_to_country(curr_country, arr_sports[count], buf);
    //додає країну до файлу спорту
    append_to_sport_file(arr_sports[count], curr_country->name);
    //оновлює файл країн
    update_countries_file(curr_country->name, arr_sports[count]);
    //оновлює кількість команд в основному списку
    sport *temp_sport = search_sport(*sport_head, arr_sports[count]);

```

```

        temp_sport->num_teams++;
    }
    free(temp);
}

```

//додає нову країну

```

void Add_country(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length)
{

```

```

    char temp[temp_length];
    int buf = 0;

```

```

    country *new_country = (country *)allocate_memory(sizeof(country));
    new_country->sports_list = NULL;
    new_country->next = NULL;

```

```

    printf("Enter the name of the new country:\n");
    gets_temp(temp);
    strcpy(new_country->name, temp);

```

//додає країну з врахуванням впорядкування за алфавітом

```

if (*country_head == NULL || strcmp(new_country->name, (*country_head)->name) < 0)

```

```

{
    new_country->next = *country_head;
    *country_head = new_country;
}

```

```

else
{

```

```

    country *current = *country_head;
    while (current->next != NULL && strcmp(current->next->name, new_country->name) < 0)
    {
        current = current->next;
    }
    new_country->next = current->next;
    current->next = new_country;
}

```

//додає спорти до листу спортів країни

```

while (1)
{
    printf("Do you want to add a sport to '%s'? (1 for Yes, 0 for No):\n", new_country->name);
    buf = get_number_in_range (0, 1);
    if (buf == 0)
        break;

    printf("Available sports:\n");
    for (int i = 0; i < *arr_length; i++)
    {

```



```

    printf("%d. %s\n", i + 1, arr_sports[i]);
}

printf("Enter the number of the sport:\n");
buf = get_number_in_range (1, *arr_length) - 1;

//додає спорт до листу спортів країни
add_sport_to_country(new_country, arr_sports[buf], 1);
//додає назву країни в файл спорту
append_to_sport_file(arr_sports[buf], new_country->name);
//знаходимо спорт в листі спортів та збільшуємо кількість команд на один
sport *sport_curr = search_sport(*sport_head, arr_sports[buf]);
sport_curr->num_teams++;

}

//оновлюємо файл країн
country_sl *sport_list = new_country->sports_list;
while (sport_list != NULL)
{
    update_countries_file(new_country->name, sport_list->sport_name);
    sport_list = sport_list->next_sport;
}
}

//додає новий спорт
void Add_sports(sport **sport_head, country **country_head, char arr_sports[][temp_length], int
*arr_length)
{
    char temp[temp_length];
    printf("Enter the name of the new sport: ");
    gets_temp(temp);

    //перевіряє що спорт ще не існує в листі
    for (int i = 0; i < *arr_length; i++)
    {
        if (strcmp(arr_sports[i], temp) == 0)
        {
            printf("The sport '%s' already exists in the list.\n", temp);
            return;
        }
    }
}

//додає новий спорт до листу спортів
sport *new_sport = (sport *)allocate_memory(sizeof(sport));
strcpy(new_sport->name, temp);
new_sport->num_teams = 1;
new_sport->next = NULL;

```

```

if (*sport_head == NULL)
{
    *sport_head = new_sport;
}
else
{
    sport *current = *sport_head;
    while (current->next != NULL)
    {
        current = current->next;
    }
    current->next = new_sport;
}

//додає новий спорт до масиву з назвами спортив
strcpy(arr_sports[*arr_length], temp);
(*arr_length)++;

//додає країну яка приймає участь в цьому спорті
char country_name[temp_length];
country *curr_country;

while (1)
{
    printf("Enter the name of a country that participates in this new sport: ");
    gets_temp(country_name);

    //перевіряє що країна існує
    curr_country = search_country(*country_head, country_name);
    if (curr_country != NULL)
    {
        printf("Country '%s' will participate in the new sport.\n", country_name);
        add_sport_to_country(curr_country, temp, 1);
        update_countries_file(curr_country->name, temp);
        break;
    }
    else
    {
        printf("Country '%s' not found in the country list.\n", country_name);
    }
}

//створює новий файл для цього спорту
strcat(temp, ".txt");
FILE *file = NULL;
file = open_check(file, temp, "w");
fprintf(file, "%s\n", country_name);
close_check(file);

```

```

    printf("File '%s' created successfully.\n", temp);
}

//видаляє країну
void Remove_info_countries(country **country_head, sport **sport_head, char arr_sports[][temp_length],
int arr_length)
{
    country *temp_country, *prev_country, *country_to_delete=NULL;
    sport *temp_sport;
    country_sl *prev_sl, *next_sl;
    char *temp = (char*)allocate_memory(temp_length*sizeof(char));

    if (*country_head == NULL)
    {
        printf("The country list is empty.\n");
        return;
    }

    printf("Which country would you like to remove?\n");
    gets_temp(temp);

    prev_country = *country_head;
    temp_country = (*country_head)->next;

    //проходить по листу країн знаходить країну для видалення і попередню країну
    if(strcmp(prev_country->name, temp)==0)
    {
        country_to_delete = *country_head;
    }
    else
    {
        while(temp_country!=NULL)
        {
            if(strcmp(temp_country->name, temp)==0)
            {
                country_to_delete = temp_country;
                break;
            }
            prev_country = temp_country;
            temp_country = temp_country->next;
        }
    }
    if (country_to_delete == NULL)
    {
        printf("Country not found.\n");
        free(temp);
        return;
    }
}

```

```

//видаляє лист спортив країни
//і зменшує кількість команд для кожного спорту в основному листі
prev_sl = country_to_delete->sports_list;
temp_sport = *sport_head;
while(prev_sl!=NULL)
{
    temp_sport = search_sport(*sport_head, prev_sl->sport_name);
    temp_sport->num_teams--;
    next_sl = prev_sl->next_sport;
    free(prev_sl);
    prev_sl = next_sl;
}

//видаляє країну
if(strcmp((*country_head)->name, temp)==0)
{
    *country_head = (*country_head)->next;
    free(country_to_delete);
}
else
{
    prev_country->next = country_to_delete->next;
    free(country_to_delete);
}

//оновлює файл країн і файл спортив
char file_name[temp_length];
Update_file("countries.txt", temp);
for(int i = 0; i < arr_length; i++)
{
    strcpy(file_name, arr_sports[i]);
    strcat(file_name, ".txt");
    Update_file(file_name, temp);
}

free(temp);
}

//видаляє спорт
void Remove_info_sports(country **country_head, sport **sport_head, char arr_sports[][temp_length], int
*arr_length)
{
    sport *sport_curr, *sport_prev;
    country *country_curr;
    country_sl *prev_sl, *next_sl;
    char *temp = (char*)allocate_memory(temp_length*sizeof(char));
    int delete_check = 0;

    if (*sport_head == NULL)

```

```

{
    printf("The sport list is empty.\n");
    return;
}

```

```

printf("Which sport would you like to remove?\n");
gets_temp(temp);

```

```

//прибирає назву спорту з масиву
for(int i = 0; i<*arr_length; i++)
{
    if(strcmp(arr_sports[i], temp)==0)
    {
        if(i!=*arr_length-1)
        {
            strcpy(arr_sports[i], arr_sports[*arr_length-1]);
            (*arr_length)--;
            delete_check = 1;
            break;
        }
        else
        {
            (*arr_length)--;
            delete_check = 1;
            break;
        }
    }
}
}

```

```

//якщо не знаходить назву, виходить
if(delete_check == 0)
{
    printf("The sport isn't found.\n");
    return;
}

```

```

sport_prev = *sport_head;
sport_curr = (*sport_head)->next;

```

```

//знаходить в основному листі спортів і видаляє
if(strcmp(sport_prev->name, temp)==0)
{
    *sport_head = (*sport_head)->next;
}
else
{
    while(sport_curr!=NULL)
    {

```

```

        if(strcmp(sport_curr->name, temp)==0)
        {
            sport_prev->next = sport_curr->next;
            free(sport_curr);
            break;
        }
        sport_prev = sport_curr;
        sport_curr = sport_curr->next;
    }
}

```

```

country_curr = *country_head;

```

```

//проходить по листу країн і видаляє з кожного списку спортів країни
while(country_curr!=NULL)

```

```

{
    prev_sl = country_curr->sports_list;
    next_sl = prev_sl->next_sport;

    if(strcmp(prev_sl->sport_name, temp)==0)
    {
        country_curr->sports_list = country_curr->sports_list->next_sport;
        free(prev_sl);
    }
    else
    {
        while(next_sl!=NULL)
        {
            if(strcmp(next_sl->sport_name, temp)==0)
            {
                prev_sl->next_sport = next_sl->next_sport;
                free(next_sl);
                break;
            }
            prev_sl = next_sl;
            next_sl = next_sl->next_sport;
        }
    }
    country_curr = country_curr->next;
}

```

```

//видаляє файл цього спорту
char file_name[temp_length];
strcpy(file_name, temp);
strcat(file_name, ".txt");
remove(file_name);

```

```

//оновлює файл "countries.txt"
FILE *input_file = NULL;
input_file = open_check(input_file, "countries.txt", "r");
FILE *temp_file = NULL;
temp_file = open_check(temp_file, "temp.txt", "w");

char *temp2 = (char*)allocate_memory(temp_length*sizeof(char));

//видаляє тільки спорт, решту строки переписує
while (fgets(temp2, temp_length, input_file))
{
    char *country_name = strtok(temp2, ":");
    char *sports = strtok(NULL, "\n");

    if (country_name == NULL || sports == NULL)
        continue;

    fprintf(temp_file, "%s: ", country_name);
    //розділяє назви спортив по комі
    char *word = strtok(sports, ", ");
    //для того щоб обробляти кому перед кожним словом
    int first_sport = 1;
    while (word != NULL)
    {
        if (strcmp(word, temp) != 0)
        {
            if (!first_sport)
                fprintf(temp_file, ", ");
            fprintf(temp_file, "%s", word);
            first_sport = 0;
        }
        word = strtok(NULL, ", ");
    }
    fprintf(temp_file, "\n");
}

free(temp);
free(temp2);
close_check(input_file);
close_check(temp_file);
remove("countries.txt");
rename("temp.txt", "countries.txt");
}

//переписує файл видаляючи згадки про країну
void Update_file(const char *filename, const char *country_to_remove)
{

```

```

FILE *file = NULL;
file = open_check(file, filename, "r");

FILE *temp_file = NULL;
temp_file = open_check(temp_file, "temp.txt", "w");

char line[256];
while (fgets(line, sizeof(line), file))
{
    //перевіряє чи є згадка країни
    if (strstr(line, country_to_remove) == NULL)
    {
        //якщо ні переписує в новий файл
        fputs(line, temp_file);
    }
}

close_check(file);
close_check(temp_file);

//замінює файл новим
remove(filename);
rename("temp.txt", filename);
}

//додає нову країну до файлу спортів
void append_to_sport_file(const char *sport_name, const char *country_name)
{
    char file_name[temp_length];
    strcpy(file_name, sport_name);
    strcat(file_name, ".txt");
    FILE *sport_file = NULL;
    sport_file = open_check(sport_file, file_name, "a");

    if (sport_file != NULL)
    {
        fprintf(sport_file, "\n%s", country_name);
        close_check(sport_file);
    }
    else
    {
        printf("Error updating file for sport %s.\n", sport_name);
    }
}

//додає новий спорт в файл countries.txt
void update_countries_file(const char *country_name, const char *new_sport)

```



```

{
    FILE *file = NULL;
    file = open_check(file, "countries.txt", "r");

    char **lines = NULL;
    int line_count = 0;
    size_t buffer_size = 1024;
    char buffer[buffer_size];

    //читає усі строки з файлу
    while (fgets(buffer, buffer_size, file))
    {
        lines = realloc(lines, (line_count + 1) * sizeof(char *));
        lines[line_count] = strdup(buffer);
        line_count++;
    }
    close_check(file);

    //оновлює строку з необхідною країною
    int found = 0;
    for (int i = 0; i < line_count; i++) {
        //перевіряє чи строка починається з назви країни
        if (strncmp(lines[i], country_name, strlen(country_name)) == 0 && lines[i][strlen(country_name)] ==
':')
        {
            found = 1;
            //прибирає \n з кінця строки
            lines[i][strcspn(lines[i], "\n")] = '\0';

            //дописує новий спорт
            strcat(lines[i], ", ");
            strcat(lines[i], new_sport);
            strcat(lines[i], "\n");

            break;
        }
    }

    //якщо не знайшли країну, додаємо нову
    if (!found)
    {
        lines = realloc(lines, (line_count + 1) * sizeof(char *));
        lines[line_count] = malloc(buffer_size * sizeof(char));
        sprintf(lines[line_count], buffer_size, "\n%s: %s", country_name, new_sport);
        line_count++;
    }

    //перепишемо файл
    file = open_check(file, "countries.txt", "w");

```

```

for (int i = 0; i < line_count; i++)
{
    fprintf(file, "%s", lines[i]);
    free(lines[i]);
}
free(lines);
close_check(file);
}

```

//приймає строку

```
void gets_temp(char *temp)
```

```

{
    while (1)
    {
        //перевіряє, що fgets щось отримав
        if((fgets(temp, temp_length, stdin))==NULL)
            printf("Error, please try again.\n");
        else
        {
            //перевіряє, що користувач не натиснув тільки Enter
            if(*temp==10)
                printf("You pressed enter only, please try again.\n");
            else
            {
                *(temp + strlen(temp)-1) = '\0';
                int i = 0;
                int bad_input = 0;
                //перевіряє що кожний символ - літера, пробел або тіре
                while(*(temp+i)!='\0')
                {
                    // 32 - space, 45 - hyphen(-), from 65 to 90 - uppercase letters, from 97 to 122 - lowercase
letters
                    if
(* (temp+i)==32||*(temp+i)==45||(* (temp+i)>=65&&*(temp+i)<=90)||(* (temp+i)>=97&&*(temp+i)<=122)))
                        i++;
                    else
                    {
                        bad_input = 1;
                        break;
                    }
                }

                if(bad_input==1)
                    printf("Please, try again. Use letters, space( ), apostrophe(') or hyphen(-) only.\n");
                else
                    break;
            }
        }
    }
}

```

```

    }

}

}

//перевіряє чи файл відкрився
FILE *open_check(FILE *fp, char str[30], char ch[3])
{
    fp = fopen(str, ch);
    if (fp==NULL)
    {
        printf("Unable to open file %s succesfully\n", str);
    }
    return fp;
}

//перевіряє чи файл закрится
void close_check(FILE *fp)
{
    if (fclose(fp)!=0)
    {
        printf("Unable to close file succesfully\n");
    }
}

//перевіряє, що користувач ввів саме число, і що воно попадає між заданих чисел
int get_number_in_range(int left, int right)
{
    int var = 0;
    int bad_input = 0;
    int i = 0;
    int negative = 0;

    while(1)
    {
        var = 0, bad_input = 0, i = 0, negative = 0;
        char number[temp_length];

        if (fgets(number, sizeof(number), stdin) == NULL) {
            printf("Error reading input. Please try again.\n");
            clearerr(stdin);
            continue;
        }

        //перевіряє, що користувач не натиснув тільки Enter
        if(number[0]==10)
        {

```

```

    bad_input = 1;
}

//перевіряє, що користувач не ввів букви, або спец знаки.
while(number[i]!=' '&&number[i]!='\n'&&number[i]!='\0')
{
    if ((number[i] - '0') < 0 || (number[i] - '0') > 9)
    {
        if(i==0&&number[i]=='-')
        {
            negative = 1;
            i++;
        }
        else
        {
            bad_input = 1;
            break;
        }
    }
    else
    {
        //якщо це не буква або знак, переводить у число
        var = var*10 + (number[i] - '0');
        i++;
    }
}

if(bad_input==1)
    printf("Please, try again, only numbers.\n");
else
{
    //якщо не попадає в рамки, просить ввести знову
    if(left==right && var != right)
        printf("Please, try again and beware of the range.\n");
    else
    {
        if(var < left || var > right)
            printf("Please, try again and beware of the range.\n");
        else
            //якщо попадає в задані рамки виход з циклу
            break;
    }
}
}

if(negative==1)
    return -var;
else
    return var;

```

```
}
```

```
//виділяє пам'ять
```

```
void* allocate_memory(size_t size)
```

```
{
```

```
    void *ptr = malloc(size);
```

```
    if (ptr == NULL) {
```

```
        printf("Memory allocation failed.\n");
```

```
        return NULL;
```

```
    }
```

```
    return ptr;
```

```
}
```

```
//звільнює лист спортив
```

```
void free_sport_list(sport *sports_head)
```

```
{
```

```
    while (sports_head) {
```

```
        sport *temp = sports_head;
```

```
        sports_head = sports_head->next;
```

```
        free(temp);
```

```
    }
```

```
}
```

```
//звільнює лист країн та лист спортив країни
```

```
void free_country_list(country *countries_head)
```

```
{
```

```
    while (countries_head)
```

```
    {
```

```
        country *temp_country = countries_head;
```

```
        countries_head = countries_head->next;
```

```
        country_sl *sports_list = temp_country->sports_list;
```

```
        while (sports_list)
```

```
        {
```

```
            country_sl *temp_sport = sports_list;
```

```
            sports_list = sports_list->next_sport;
```

```
            free(temp_sport);
```

```
        }
```

```
        free(temp_country);
```

```
    }
```

```
}
```