

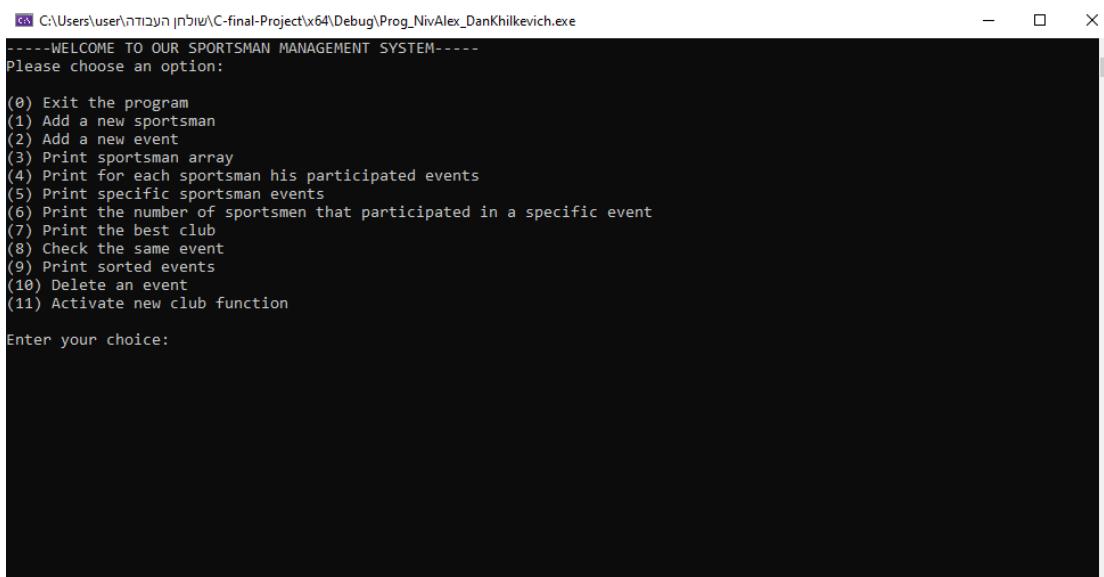
C Programming Language - Intro to Computer Science



מגישיים:

ניב אלכס, מייל: nivalex.01@gmail.com
דן חילקביץ', מייל: danchuk213@gmail.com
הערות להתמצאות במערכת הספורטאים:

זה מסך הפלט הראשי של המערכת שלנו, הוא מציע למשתמש 11 אופציות של בחירה לפעולות במערכת



```
C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhalkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function
Enter your choice:
```

לחיצה על 0 – יצא מהמערכת

**לחיצה על 1 – הוספה ספורטאי חדש למערכת (מפעילה את הפונקציה
(addSportman)**

**לחיצה על 2 – הוספה אירוע חדש למערכת (מפעילה את הפונקציה
(addEvent)**

**לחיצה על 3 – מדפסה את מידע הספורטאים הנוכחי (מקבלת אינפורמציה זו
מהקובץ "SportsmanData.txt")**

**לחיצה על 4 – מדפסה עבור כל ספורטאי במערכת הספורטאים את האירועים
השייכים לו**

**לחיצה על 5 – המשתמש יזין את שם המשפחה של ספורטאי ספציפי והמערכת
תדפיס את האירועים של אותו ספורטאי.**

לחיצה על 6 – המשתמש יזין פרט אירוע מסוים (שם אירוע ורשות) והמערכת תדפיס את מספר הספורטאים שהשתתפו באותו אירוע (מפעילה את הפונקציה CountEvent)

לחיצה על 7 – מדפיסה את שם המועדון הטוב ביותר (BestClub). המועדון הטוב ביותר ביותר הוא המועדון שיש לו את הספורטאים שהשתתפו במספר הרב ביותר של אירועים (בנוסף לפונקציה מדפיסה את מספר האירועים בהם ספורטאי המועדון הטוב ביותר השתתף)

לחיצה על 8 – המשתמש יזין תעודת זהות של ספורטאי מסוים. המערכת תדפיס את שמות כל הספורטאים שהשתתפו באותו אירוע יחד עם אותו ספורטאי (ההדפסה תהיה שם אירוע ומילוי שהשתתף בלבד עם אותו ספורטאי באירוע זה, הפעלה של הפונקציה CheckSameEvent)

לחיצה על 9 – המשתמש יזין שם של מועדון ספורט מסוים. המערכת תדפיס בצורה ממוקנת ("עפ"י שנה") את האירועים של כל הספורטאים באותו המועדון (הפעלה של הפונקציה PrintSortedEvents)

לחיצה על 10 – המשתמש יזין אירוע (שם ורשות) אותו הוא רוצה למחוק מהמערכת. המערכת תמחק אירוע זה ותעדכן את הנתונים בהתאם (הפעלה של הפונקציה Delete event)

לחיצה על 11 – המשתמש יזין שמות של 2 מועדוני ספורט, המערכת תיצור קובץ חדש בשם "Club.txt" בו יהיו נתונים שמות כל האירועים בהם השתתפו הספורטאים משני המועדונים (ללא חזרות), (הפעלת הפונקציה NewClub)

לחיצה על כל מסטר אחר – המערכת תזהה כי מדובר בקלט לא תקין ותבקש מהמשתמש להזין שוב מספר בין 0 ל-11 כולל.

- במהלך כתיבת הקוד הקפדנו על כתיבת העורות שיעזרו למי שקורא את העבודה להבין את דרך החשיבה שלנו ואת דרך המימוש שלנו. לפני כל פונקציה כתבנו בקצרה מה היא מקבלת ומה היא עשויה/מחזירה.

- קבועים: במהלך התוכנית עצמה הינו צריכים להתעסק מספר פעמים עם משתנים אשר יכולים להיות קבועים. המשתנים הקבועים בהם השתמשנו הם `MAX_LINE_LENGTH` שמצוין את אורך התווים המקסימלי אשר יכול להיות בשורה (הגדרנו זאת ל-100). `MAX_NAME_LENGTH` מצוין את אורך השם המקסימלי (יכול להיות שם פרטי, שם משפחה, שם קבוצה, שם אירוע וכו'...) אותו הגדרנו להיות עד 50 תוים. `MAX_CLUB_NUMBER` מדווח המעודדים המקסימלי את התוכנית לקבל (הגדרנו מספר זה להיות גבוהה מאוד, 200, בהנחה שבתוכנית שלנו יהיו לפחות 200 מועדוני ספורט). `MAX_EVENT_NUMBER` מספר האירועים המקסימלי שיכול להיות בתוכנית שלנו (הגדרנו זאת גם כ-200). חשוב לציין שמאחר ועבדנו עם קבועים ניתן להתאים את התוכנית שלנו בקלות ולשנות בקלות יתרה את גודלם של קבועים אלה לגודלים/קטנים יותר.

- אופן ההתייחסות למערך הספורטאים ומערך האירועים – כפי שתכתוב בהנחיות העבודה, מערך הספורטאים הינו מערך דינמי (גודלו ותוכנו יכול לשינויים במהלך ריצת התוכנית). לכן הקפדנו בפונקציות שעושות שימוש במערך הספורטאים לעבוד עם פוינטרים. זאת הסיבה שפעמים רבות בתוכנית שלנו אנחנו שולחים לפונקציות מצביע למערך הספורטאים ומצביע לגודל מערך הספורטאים (מתוך ההנחה שהגדלים אלו משתנים במהלך התוכנית). לכל ספורטאי יש גם מערך דינמי של אירועים, לכן גם במסגרת העבודה עם מערך האירועים הקפדנו לעבוד עם פוינטרים.

- הקצאה ושחרור של זיכרון – כפי שהוגדר בהוראות העבודה, מערך הספורטאים הינו דינמי וכך גם מערך האירועים של כל ספורטאי. לכן בסיום התוכנית הקפדנו לשחרר זיכרון עבור מערך הספורטאים ועובדך מערך האירועים של כל ספורטאי. בנוסף, שם המעודן של כל ספורטאי, שם האירוע ומקום האירוע (`p2club,p2title,p2location`) הוגדרו בעבודה כמצביים ל-`char`. משפט גודלם של מחוזות אלה הוא דינמי. לכן הקפדנו לבדוק גם מקרים אלה.

- הסבר על הפונקציה `strcpy` שהשתמשנו בה רבות מהלך העבודה – פונקציה זו (שהכרנו גם במהלך התרגולים) מעתקה את הערך של מחרוזת מסוימת ולכתובת מסוימת אותה היא מקבלת. עשינו שימוש בפונקציה זו פעמים רבות, זאת על מנת לעשות השמה של ערך (value) לכתובת מסוימת.
- הסבר על בדיקות ותקינות הקלט – במהלך העבודה הקפదנו על ביצוע מספר רב של בדיקות ותקינות קלט. הקפదנו לבדוק שבעת ה הכנסת ספורטאי חדש למערכת, תעודת הזהות שלו תהיה רק ערך מספרי, השם פרטי ושם המשפחה שלו יהיה רק ערך של מחרוזת והמין שלו יהיה רק 0 או 1 (כפי שהוגדר בשאלה). אם המשתמש ינסה להכניס ספורטאי חדש עם ערכים לא תקינים המערכת תודיע לו לגבי זה. לגבי שמות המועודונים, גם התרנו למשתמש להכניס גם מספרים מאחר ובעולם האמתי קיימים מועודוני ספורט שבשם יש מספרים (כמו שאלה 04, או הנובר 96 שאלות מועודוני ספורט אמתיים מגרמניה).
- הסבר על פורמט הקבצים ואופן הטיפול בהכנסת ספורטאי/אירוע חדש – לאחר ובקבצי הספורטאים והאירועים הפורמט הוא צזה שככל מילה מתחילה באות גזולה, הקפదנו להגדיל את האות הראשונה של כל מילה שנקלטה מהמשתמש ובכך לשמר על קונסיסטנטיות מערכת שלנו. גם למשל אם יוכנס ספורטאי חדש בשם: alex זה הוא ייכתב לקובץ programmers race שם programmers race הוא ייכתב לקובץ האירועים כר: programmers Race .Alex Niv. ואם יוכנס אירוע אירוע כר: programmers race .Programmers Race

```

program.c: 1 //Prop_NonSec_DataSource.h
2
3 #define _CRT_SECURE_NO_WARNINGS
4
5 #define MAX_LINE_LENGTH 100 //we assume the maximum length of line can be 100 characters;
6 #define MAX_NAME_LENGTH 50 //we assume the maximum length of name (club_name or first_name or last_name) can be up to 100 characters;
7 #define MAX_CLUB_NUMBER 200 //we assume the maximum number of clubs can be 200;
8 #define MAX_EVENT_NUMBER 200 //we assume the maximum number of events can be 200;
9
10 //event structure
11 typedef struct
12 {
13     char* p2title; //event name
14     char* p2location; //event location
15     int year; //event year
16 } event;
17
18 //sportsman structure
19 typedef struct {
20     int id; // id of sportsman
21     char Fname[MAX_NAME_LENGTH]; //first name
22     char Lname[MAX_NAME_LENGTH]; // last name
23     char* p2club; //pointer to sportsman club
24     int gen; //gender
25     event* p2events; //pointer to a dynamic arr of events
26     int Nevents; //number of participated events
27 } sportsman;
28
29 //clearInputBuffer function clear the buffers for each word (spaces)
30 void clearInputBuffer()
31 {
32     int c;
33     while ((c = getchar()) != '\n' && c != EOF);
34 }
35
36 //printSportsmen function gets array of sportsmen and number of sportsmen
37 // the function prints the details(id,first name,last name, club, gender, Nevents, P2events) for each sportsmen
38 void printSportsmen(sportsman* sportsmen_array, int *num_sportsmen)
39 {
40     for (int i = 0; i < *num_sportsmen; ++i) {
41         printf("Sportsman ID %d:\n", sportsmen_array[i].id);
42         printf("First Name: %s\n", sportsmen_array[i].Fname);
43         printf("Last Name: %s\n", sportsmen_array[i].Lname);
44
45         // check if p2club is not NULL before printing
46         if (sportsmen_array[i].p2club != NULL) {
47             printf("Club: %s\n", sportsmen_array[i].p2club);
48         }
49         else {
50             printf("Club: Not specified\n");
51         }
52
53         printf("Gender: %d\n", sportsmen_array[i].gen);
54         printf("Number of Events: %d\n", sportsmen_array[i].Nevents);
55         printf("Events:\n");
56
57         // check if p2events is not NULL before accessing
58         if (sportsmen_array[i].p2events != NULL) {
59             for (int j = 0; j < sportsmen_array[i].Nevents; ++j) {
60                 printf("- Event %d: %s, %s, %d\n", j + 1, sportsmen_array[i].p2events[j].p2title,
61                     sportsmen_array[i].p2events[j].p2location, sportsmen_array[i].p2events[j].year);
62             }
63         }
64         else {
65             printf("No events registered\n");
66         }
67     }
68 }
69
70 //print_Sportman_participated_events gets array of sportsmen and number of sportsmen
71 // the function prints for each sports men the events that he is participated.
72 void print_Sportman_participated_events(sportsman* sportsmen_array, int *num_sportsmen)
73 {
74     for (int sportman_index = 0; sportman_index < (*num_sportsmen); sportman_index++) {
75         printf("%s %s:\n", sportsmen_array[sportman_index].Fname, sportsmen_array[sportman_index].Lname);
76
77         // Check if the sportsman has any events
78         if (sportsmen_array[sportman_index].Nevents == 0 || sportsmen_array[sportman_index].p2events == NULL) {
79             printf("No events participated.\n");
80             continue; // Move to the next sportsman
81         }
82
83         // Print each event the sportsman participated in
84         for (int event_index = 0; event_index < sportsmen_array[sportman_index].Nevents; event_index++) {
85             printf("- %s, %s, %d\n",
86                 sportsmen_array[sportman_index].p2events[event_index].p2title,
87                 sportsmen_array[sportman_index].p2events[event_index].p2location,
88                 sportsmen_array[sportman_index].p2events[event_index].year);
89         }
90     }
91 }
92
93 //countLines function gets FileName and return the number of lines (without count the first line)
94 int countLines(const char* filename)
95 {
96     FILE* file_ptr = fopen(filename, "r");
97     int number_of_lines = 0; // we don't want to count the first line of the file.
98     int ch;
99     char line[MAX_LINE_LENGTH];
100
101    if (file_ptr == NULL)
102    {
103        printf("Error, can not opening the file\n");
104        return -1;
105    }
106
107    while ((ch = fgetc(file_ptr)) != EOF)
108    {
109        if (ch == '\n')
110        {
111            {
112                number_of_lines++;
113            }
114        }
115    }
116    fclose(file_ptr);
117    return number_of_lines;
118 }

```

FromFile2Sportsman.c

```

118 //FromFile2Sportsman reads data from "SportsmenData.txt" File and create new sportsmen_array with this data.
119 //the function gets a pointer to sportsmen_array and the sportsmen_array_size and because it "void" function we worked by reference.
120 void FromFile2Sportsman(const char* filename, sportsman** sportsmen_array, int sportsmen_array_size)
{
    char line[MAX_LINE_LENGTH];
    int sportsmen_count = 0;
    FILE* file = fopen(filename, "r");
    if (file == NULL)
    {
        printf("Error opening file %s\n", filename);
        exit(1);
    }
    sportsmen_array = malloc(sportsmen_array_size * sizeof(sportsman)); // allocate memory for sportsmen array;
    if (*sportsmen_array == NULL) // allocate memory checking
    {
        printf("Memory allocation failed\n");
        exit(1);
    }
    fgets(line, sizeof(line), file); // skip the format line (the first line)
    while (fgets(line, sizeof(line), file) != NULL && sportsmen_count < sportsmen_array_size) // read line by line from the file
    {
        int gen;
        char clubName[100]; // temporary buffer for club name, club name = maximum of 100 chars
        if (sscanf(line, "%d[%c];%c[%c];%d", &(*sportsmen_array)[sportsmen_count].id,
                   (*sportsmen_array)[sportsmen_count].Name,
                   (*sportsmen_array)[sportsmen_count].Name, &gen) != 5)
        {
            printf("Error reading data from file\n");
            exit(1);
        }
        sportsmen_array[sportsmen_count].gen = gen;
        (*sportsmen_array)[sportsmen_count].Nevents = 0;
        (*sportsmen_array)[sportsmen_count].p2events = NULL;
        // Dynamically allocate memory to p2club based on the length of clubName
        (*sportsmen_array)[sportsmen_count].p2club = malloc((strlen(clubName) + 1) * sizeof(char));
        if ((*sportsmen_array)[sportsmen_count].p2club == NULL)
        {
            printf("Memory allocation failed\n");
            exit(1);
        }
        strcpy((*sportsmen_array)[sportsmen_count].p2club, clubName);
        sportsmen_count++;
    }
    fclose(file);
}

```

FromFile2Events.c

```

169 //FromFile2Events reads data from "EventData.txt" File and create new events_array with this data.
170 //the function gets a pointer to sportsmen_array and the sportsmen_array_size and because it "void" function we worked by reference.
171 void FromFile2Events(const char* filename, sportsman* sportsmen_array, int num_sportsmen)
{
    FILE* file = fopen(filename, "r");
    if (file == NULL)
    {
        printf("Error can not opening the file %s\n", filename);
        return;
    }
    char line[MAX_LINE_LENGTH]; // assuming maximum line length is 100 characters
    int sportsman_id, year;
    char event_name[MAX_NAME_LENGTH], location[MAX_NAME_LENGTH];
    while (fgets(line, sizeof(line), file) != NULL) // read each line to get [sportsman id, event name, location, and year]
    {
        sscanf(line, "%d[%c,%c,%c,%c,%d", &sportsman_array[sportsmen_id].id, &event_name[0], &event_name[1], &event_name[2], &event_name[3], &year);
        sportsman* current_sportsman = NULL;
        for (int i = 0; i < num_sportsmen; i++)
        {
            if (sportsmen_array[i].id == sportsman_id) // Found the matching sportman in the sportman array
            {
                current_sportsman = &sportsmen_array[i];
                break;
            }
        }
        // we know that sportman is found, now we add the event to their list of events
        if (current_sportsman != NULL)
        {
            // Relocate memory for the events array (event arr is daynamic array)
            current_sportsman->p2events = realloc(current_sportsman->p2events, (current_sportsman->Nevents + 1) * sizeof(event));
            if (current_sportsman->p2events == NULL)
            {
                printf("Memory allocation failed.\n");
                exit(1);
            }
            // Assign event details
            event* current_event = &(current_sportsman->p2events[current_sportsman->Nevents]);
            current_event->p2title = (char*)malloc(strlen(event_name) + 1); //the title of the event is dynamic
            if (current_event->p2title == NULL)
            {
                printf("Memory allocation failed.\n");
                exit(1);
            }
            strcpy(current_event->p2title, event_name); //replace the address of p2title with value (the title)

            current_event->p2location = (char*)malloc(strlen(location) + 1); //the location of the event is dynamic
            if (current_event->p2location == NULL)
            {
                printf("Memory allocation failed.\n");
                exit(1);
            }
            strcpy(current_event->p2location, location); //replace address of p2location with value (the location)
            current_event->year = year; // set value
            current_sportsman->Nevents++; // add 1 to the number of events for this specific sportman.
        }
    }
    fclose(file); //close EventData.txt file
}

```

```

230 //check_if_sportsman_already_exists function gets pointer to sportsmen_array, size,new_sportman_id and check if this sportman
231 // already exists in the sportsmen_array. If it exists the function return 0 and if not the function return 1.
232 int check_if_sportsman_already_exists(sportsman* sportsmen_array[], int new_sportman_id, int sportsmen_array_size)
233 {
234     // Check if the sportsman already exists in the array
235     for (int i = 0; i < sportsmen_array_size; i++)
236     {
237         if ((*sportsmen_array)[i].id == new_sportman_id)
238         {
239             printf("Error: Sportsman with ID %d already exists.\n", new_sportman_id);
240             return 0; // The sportsman already exists in the array
241         }
242     }
243     return 1; // The sportsman does not exist in the array
244 }
245
246 //capitalize_each_First_Letter get a string and capitalize the first letter of each word
247 void capitalize_each_First_Letter(char* str)
248 {
249     int i;
250
251     // Capitalize first letter of the string
252     if (str[0] != '\0') {
253         str[0] = toupper(str[0]);
254     }
255
256     // Capitalize first letter of each word
257     for (i = 1; str[i] != '\0'; i++) {
258         if (isspace(str[i - 1]) && isalpha(str[i])) {
259             str[i] = toupper(str[i]);
260         }
261     }
262 }
263

```

```

264
265 int is_number(const char* str) {
266     // check if each character is a digit
267     for (int i = 0; str[i] != '\0'; i++) {
268         // Check if the character is not between '0' and '9'
269         if (str[i] < '0' || str[i] > '9')
270         {
271             return 1; //the str is not a number;
272         }
273     }
274     return 0; //the str is number
275 }
276

```

```

277 //addSportsman function gets pointer to sportsmen_arr and pointer to sportsmen_array_size
278 // the function add new sportmen to the array and change his size.
279 // note: sportsmen_array is a dynamic array and sportsmen_array_size is a dynamic size so we worked with pointers.
280 int addSportsman(sportsman* sportsmen_array[], int* sportsmen_array_size)
281 {
282     int new_sportman_id;
283     char club_name[MAX_NAME_LENGTH];
284
285     printf("Please enter the new sportsman details:\n");
286     while (1)
287     {
288         printf("Enter the new sportsman ID: ");
289         if (scanf("%d", &new_sportman_id) != 1)
290         {
291             printf("Error: Invalid input for ID. Please enter a number.\n");
292             while (getchar() != '\n'); // Clear input buffer
293         }
294         else
295         {
296             break; // Exit the loop if a valid ID is provided
297         }
298     }

```

```

299
300     if (check_if_sportsman_already_exists(sportsmen_array, new_sportman_id, *sportsmen_array_size) == 0) // Check if the sportsman already exists
301         return 0; // The sportsman already exists in the array, return 0
302
303     (*sportsmen_array_size)++; // add +1 to the size of the sportsmen array
304
305     sportsman* temp = realloc(sportsmen_array, (*sportsmen_array_size) * sizeof(sportsman)); // realloc memory for the new sportsmen array
306     if (temp == NULL)
307     {
308         printf("Error: Memory allocation failed.\n");
309         return 0;
310     }
311     *sportsmen_array = temp;
312     char first_name[MAX_NAME_LENGTH];
313     char last_name[MAX_NAME_LENGTH];
314     //char club_name[MAX_NAME_LENGTH];
315     // insert the details of the new sportsman
316     while (1)
317     {
318         printf("Enter the new sportsman First name: ");
319         scanf("%s", first_name);
320
321         if (is_number(first_name)==0)
322         {
323             printf("You entered a number instead of a name. Please try again.\n");
324         }
325         else
326         {
327             // Copy the first name to the Fname field of the last sportsman in the array
328             strcpy(&sportsmen_array)[*sportsmen_array_size - 1].Fname, first_name);
329             break;
330         }
331     }
332     capitalize_each_First_Letter(&sportsmen_array)[*sportsmen_array_size - 1].Fname); // capitalize first letter of each word
333

```

```

335     while (1)
336     {
337         printf("Enter the new sportsman Last name: ");
338         scanf("%s", last_name);
339
340         if (is_number(last_name) == 0)
341         {
342             printf("You entered a number instead of a name. Please try again.\n");
343         }
344         else
345         {
346             // Copy the first name to the Name field of the last sportsman in the array
347             strcpy((*sportsmen_array)[*sportsmen_array_size - 1].Lname, last_name);
348             break;
349         }
350     }
351
352     capitalize_each_First_Letter((*sportsmen_array)[*sportsmen_array_size - 1].Lname); // capitalize first letter of each word
353     printf("Enter the new sportsman Club name: ");
354     getch();
355     gets(club_name, sizeof(club_name), stdin);
356     club_name[strlen(club_name) - 1] = '\0'; // Remove trailing newline character
357     capitalize_each_First_Letter(club_name);
358     // malloc memory for the club name and copy the string instead of address,
359     (*sportsmen_array)[*sportsmen_array_size - 1].p2club = malloc(strlen(club_name) + 1) * sizeof(char);
360     if ((*sportsmen_array)[*sportsmen_array_size - 1].p2club == NULL)
361     {
362         printf("Error: Memory allocation failed.\n");
363         return 0;
364     }
365     strcpy((*sportsmen_array)[*sportsmen_array_size - 1].p2club, club_name);
366     int gender_input;

```

```

367     do
368     {
369         printf("Enter the new sportsman Gender (1 for male or 0 for female): ");
370         if (scanf("%d", &gender_input) != 1) //check if the input is not a number (the input is string)
371         {
372             printf("Error: Invalid input for gender.\n"); //the user entered string instead of a number
373             while (getchar() != '\n') // Clear input buffer
374             {
375             }
376             //the user enterd number, we need to check if the number is only 1 or 0
377             else if (gender_input != 0 && gender_input != 1)
378             {
379                 printf("Error: Gender should be 0 (female) or 1 (male).\n"); //the number is not 1 or 0
380             }
381             else //valid input
382             {
383                 (*sportsmen_array)[*sportsmen_array_size - 1].gen = gender_input;
384                 break; // good input, now we can exit the loop
385             }
386         } while (1); // running until we get good input
387
388         (*sportsmen_array)[*sportsmen_array_size - 1].id = new_sportman_id;
389         (*sportsmen_array)[*sportsmen_array_size - 1].Nevents = 0;
390         (*sportsmen_array)[*sportsmen_array_size - 1].p2events = NULL;
391
392     } // successsed to add the new sportman to the array

```

```

393 //Find_Sportsman_ByID gets id number, sportsmen_array and sportsmen_array_size
394 //the function return sportman struct with the specific id it get.
395 //sportsman* find_Sportsman_ByID(int id, sportsman* sportsmen_array[], int* sportsmen_array_size)
396 {
397     if (sportsmen_array == NULL || sportsmen_array_size == NULL)
398     {
399         return NULL;
400     }
401
402     for (int i = 0; i < *sportsmen_array_size; i++)
403     {
404         sportsman* current_sportsman = sportsmen_array[i];
405         if (current_sportsman == NULL)
406             // Handle NULL pointer in the array
407             continue;
408
409         if (current_sportsman->id == id)
410         {
411             return current_sportsman; // Found, return the sportsman
412         }
413     }
414
415     return NULL; // Sportsman with the given ID not found
416
417 }

```

```

419 //isEventExist function get and array of events, number of events, event name,location and year
420 //the function return 1 if the event is already exists on the events list or 0 if its not already exists
421 int isEventExist(event* events_array, int num_events, const char* event_name, const char* location, int year)
422 {
423     char current_event_name[50];
424     char current_event_location[50];
425     int current_event_year;
426     for (int i = 0; i < num_events; i++)
427     {
428         if (events_array[i].p2title == NULL || events_array[i].p2location == NULL)
429         {
430             printf("Error: Null pointer encountered.\n");
431             continue; // move to the next event
432         }
433
434         strcpy(current_event_name, events_array[i].p2title);
435         strcpy(current_event_location, events_array[i].p2location);
436         current_event_year = events_array[i].year;
437         if (strcmp(current_event_name, event_name) == 0 &&
438             strcmp(current_event_location, location) == 0 &&
439             current_event_year == year)
440         {
441             return 1; // event already exists
442         }
443     }
444     return 0; // event does not exist
445 }
446

```

```

447 int does_string_has_number(const char* str)
448 {
449     for (int i = 0; str[i] != '\0'; i++)
450     {
451         if (str[i] > 48 && str[i] < 57) //ascii table
452         {
453             return 1; // the number found
454         }
455     }
456     return 0; // the number not found
457 }
458
459 int isString(const char* str)
460 {
461     while (*str)
462     {
463         if (!isalpha(*str) && !isspace(*str))
464         {
465             return 0; // Not a string (contains non-alphabetic characters)
466         }
467         str++;
468     }
469     return 1; // String
470 }

```

```

467 //addEvent function gets sportsman_array, sportsman_id and sportsman_array size
468 //the function add new event to this sportsman
469 void addEvent_to_club_events(sportsman* sportsmen_array, int sportsman_id, int* sportsmen_arr_size)
470 {
471     char event_name[MAX_NAME_LENGTH], location[MAX_NAME_LENGTH];
472     int year;
473     clearInputBuffer();
474     printf("Enter the new event name: ");
475     fgets(event_name, sizeof(event_name), stdin);
476     event_name[strcspn(event_name, "\n")] = '\0'; //remove the \n character and replace it with \0
477 
478     while (!isString(event_name)) {
479         printf("Error: Event name must consist of alphabetic characters only. Please try again.\n");
480         printf("Enter the new event name: ");
481         fgets(event_name, sizeof(event_name), stdin);
482         event_name[strcspn(event_name, "\n")] = '\0'; //remove the \n character and replace it with \0
483     }
484 
485     while (1)
486     {
487         printf("Enter the new event location: ");
488         fgets(location, sizeof(location), stdin);
489         location[strcspn(location, "\n")] = '\0'; //remove the \n character and replace it with \0
490 
491         if (!isString(location)) {
492             printf("Error: Location must consist of alphabetic characters only. Please try again.\n");
493         }
494         else {
495             break;
496         }
497     }
498 }
499 

500 while (1)
501 {
502     printf("Enter the event year: ");
503     if (scanf("%d", &year) != 1) {
504         printf("Error: Invalid input for event year. Please enter a number.\n");
505         clearInputBuffer(); // Clear input buffer
506     }
507     else {
508         clearInputBuffer(); // Clear input buffer
509         break; // we exit the loop if a valid year is entered
510     }
511 
512 // find the index of the sportsman in the sportsmen array
513 int sportsman_index = -1;
514 for (int i = 0; i < *sportsmen_arr_size; i++) {
515     if (sportsmen_array[i].id == sportsman_id)
516     {
517         sportsman_index = i;
518         break;
519     }
520 }
521 
522 // if the sportsman is not found, return 0 with an error
523 if (sportsman_index == -1) {
524     printf("Error: Sportsman with ID %d not found.\n", sportsman_id);
525     return 0;
526 }
527 
528 capitalize_each.First_Letter(event_name); //each event will start with capital letter in the original file
529 capitalize_each.First_Letter(location); //each location will start with capital letter in the original file
530 // we check if the event already exists
531 if (isEventExist(sportsmen_array[sportsman_index].p2events, sportsmen_array[sportsman_index].Nevents, event_name, location, year) == 1) {
532     printf("The event already exists for this sportman\n\n");
533     return 0; // Event already exists
534 }

535 // realloc memory for the events array of the sportsman (this is a dynamic array)
536 sportsmen_array[sportsman_index].p2events = realloc(sportsmen_array[sportsman_index].p2events, (sportsmen_array[sportsman_index].Nevents + 1) * sizeof(event));
537 if (sportsmen_array[sportsman_index].p2events == NULL) {
538     printf("Error: Memory allocation failed.\n\n");
539     return 0; // failed to allocate memory
540 }
541 
542 // allocate memory for the new event details
543 sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2title = malloc((strlen(event_name) + 1) * sizeof(char));
544 if (sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2title == NULL) {
545     printf("Error: Memory allocation failed.\n\n");
546     return 0;
547 }
548 
549 sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2location = malloc((strlen(location) + 1) * sizeof(char));
550 if (sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2location == NULL) {
551     printf("Error: Memory allocation failed.\n\n");
552     free(sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2title);
553     return 0;
554 }
555 
556 // copy event details
557 strcpy(sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2title, event_name);
558 strcpy(sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].p2location, location);
559 sportsmen_array[sportsman_index].p2events[sportsmen_array[sportsman_index].Nevents].year = year;
560 
561 // add +1 to this sportman number of events
562 sportsmen_array[sportsman_index].Nevents++;
563 
564 return 1; // the event added successfully so we return 1
565 }

566 }

567 //Write_array_to_Sportsman_data function gets file_name, sportsmen_array, and num_sportsmen
568 //the function writes the updated array of sportsmen to the file "SportsmanData.txt"
569 void Write_array_to_Sportsman_data(const char* filename, sportsman* sportsmen_array, int num_sportsmen)
570 {
571     FILE* file_dest = fopen("SportsmanData.txt", "w");
572     fprintf(file_dest, "%d;%s;%s;%d", sportsmen_array[0].id, sportsmen_array[0].Fname,
573             sportsmen_array[0].Lname, sportsmen_array[0].p2club,
574             sportsmen_array[0].gen);
575     if (0 != num_sportsmen - 1) //to avoid blank line in the end of the file. we want to write the last one without new line
576     {
577         fprintf(file_dest, "\n");
578     }
579     fclose(file_dest);
580     printf("Array successfully written to %s.\n", filename);
581 }
582 
```

```

588 //Write_array_to_Event_Data function gets file_name, sportsmen_array, and num_sportsmen
589 //The function writes the updated array of events to the file "EventData.txt"
590 void Write_array_to_Event_Data(const char* filename, sportsman* sportsmen_array, int num_sportsmen)
591 {
592     FILE* file_dest = fopen(filename, "w");
593     // Write each event information to the file
594     for (int sportman_index = 0; sportman_index < num_sportsmen; sportman_index++)
595     {
596         for (int event_index = 0; event_index < sportsmen_array[sportman_index].Nevents; event_index++)
597         {
598             fprintf(file_dest, "%d,%s,%s\n",
599                     sportsmen_array[sportman_index].id,
600                     sportsmen_array[sportman_index].p2events[event_index].p2title,
601                     sportsmen_array[sportman_index].p2events[event_index].p2location,
602                     sportsmen_array[sportman_index].p2events[event_index].year);
603         }
604     }
605     fclose(file_dest);
606     printf("Events successfully written to %s.\n", filename);
607 }
608
609
610 //sportsman* getSportsmanByID(int sportsman_id, sportsman* sportsmen_array, int num_sportsmen)
611 {
612     for (int i = 0; i < num_sportsmen; i++)
613     {
614         if (sportsmen_array[i].id == sportsman_id) //return a pointer to the found sportsman
615         {
616             return &sportsmen_array[i];
617         }
618     }
619     return NULL; //if not found return null
620 }
621
622 // function to remove leading and trailing whitespace characters from a string
623 void clean_char_from_spaces(char* str)
624 {
625     int start = 0, end = strlen(str) - 1;
626     while (isspace(str[start])) {
627         start++;
628     }
629     while (end > start && isspace(str[end])) {
630         end--;
631     }
632     str[end + 1] = '\0';
633     if (start > 0) {
634         // Shift the trimmed string to the beginning of the original string
635         memmove(str, str + start, end - start + 2);
636     }
637 }
638
639
640
641 // help function to clean leading and trailing spaces from a string. this is for countEvent function
642 void clean_event_name_buffer(char* str)
643 {
644     while (isspace((unsigned char)*str)) {
645         str++;
646     }
647     if (*str == '\0') {
648         return;
649     }
650     char* end = str + strlen(str) - 1;
651     while (end > str && isspace((unsigned char)*end)) {
652         end--;
653     }
654     *(end + 1) = '\0';
655 }
656
657
658 //get_sportsman_by_lastname get sportsman last name, sportsmen_array and pointer to number of sportsmen
659 //the function return struct of sportsman with this last name
660 sportsman* get_sportsman_by_lastname(char sportsman_last_name[], sportsman* sportsmen_array, int* num_sportsmen)
661 {
662     clean_char_from_spaces(sportsman_last_name);
663     capitalize_each_First_Letter(sportsman_last_name);
664     for (int i = 0; i < *num_sportsmen; i++)
665     {
666         if (strcmp(sportsman_last_name, sportsmen_array[i].Lname) == 0)
667         {
668             return &sportsmen_array[i]; // return a pointer to the found sportsman
669         }
670         else
671         {
672             continue;
673         }
674     }
675     return NULL; // return null if the sportsman with this last name is not found
676 }
677
678
679 //printEvents function gets sportsman last name, sportsmen_array and a pointer to number of sportsmen
680 //the function print for each sportsman the event which he participated in.
681 //If success we return 1
682 //If sportsman last name not found we return -1
683 //If sportsman event lis is empty we return 0
684 int printEvents(char sportsman_last_name[], sportsman* sportsmen_array, int* num_sportsmen)
685 {
686     // call help function to get sportsman struct by last name
687     sportsman* found_sportsman = get_sportsman_by_lastname(sportsman_last_name, sportsmen_array, num_sportsmen);
688
689     // If sportsman with the given last name is found
690     if (found_sportsman != NULL)
691     {
692         if (found_sportsman->p2events == NULL) // Check if the sportsman has no events
693         {
694             printf("No events found for %s.\n", found_sportsman->Fname, found_sportsman->Lname);
695             return 0; // return 0 to indicate no events found
696         }
697         else
698         {
699             printf("Events for %s:\n", found_sportsman->Fname, found_sportsman->Lname);
700             // print each event for the found sportsman
701             for (int j = 0; j < found_sportsman->Nevents; j++)
702             {
703                 printf("%s, %s, %d\n", found_sportsman->p2events[j].p2title,
704                       found_sportsman->p2events[j].p2location, found_sportsman->p2events[j].year);
705             }
706         }
707     }
708     else
709     {
710         return -1; // return -1 to indicate sportsman not found
711     }
712     return 1; // return 1 to indicate success
713 }
```

```

713 //countEvent function gets event name and year and the sportsman_array and number
714 //the function counts the number of sportsmen who participated in event E in the year Y
715 int countEvent(char* event_name, int Y, sportsman* sportsmen_array, int* num_sportsmen) {
716     int count = 0;
717     char event_name[MAX_NAME_LENGTH];
718     strcpy(event_name, E);
719     clean_event_name_buffer(event_name);
720
721     // running on each sportsman in the array
722     for (int i = 0; i < *num_sportsmen; i++) {
723         sportsman current_sportsman = sportsmen_array[i];
724
725         // check if the current sportsman has participated in the event in the specified year
726         for (int j = 0; j < current_sportsman.Nevents; j++) {
727             char event_title[MAX_NAME_LENGTH];
728             strcpy(event_title, current_sportsman.p2events[j].p2title);
729             clean_event_name_buffer(event_title);
730
731             if (strcmp(event_title, event_name) == 0 && current_sportsman.p2events[j].year == Y) {
732                 count++;
733             }
734             break; // No need to continue checking events for this specific sportsman
735         }
736     }
737
738     return count;
739 }
740
741
742 //getSportClubName gets pointer to club name (p2club) and return the name of the club
743 char* getSportClubName(char* p2club)
744 {
745     size_t length = strlen(p2club); // find the length of the club name
746
747     // allocate memory for the club name without the first and last spaces (note: club name is dynamic)
748     char* club_name = (char*)malloc(length + 1);
749     if (club_name == NULL) {
750         printf("Memory allocation failed\n");
751         exit(1);
752     }
753
754     size_t start = 0; //find the index of the first non-space character
755     while (p2club[start] == ' ') {
756         start++;
757     }
758     size_t end = length - 1; //find the index of the last non-space character
759     while (end > 0 && p2club[end] == ' ') {
760         end--;
761     }
762     size_t j = 0; //copy the club name without the first and last spaces
763     for (size_t i = start; i <= end; i++) {
764         club_name[j++] = p2club[i];
765     }
766
767     club_name[j] = '\0'; //add \0 that indicate the end of string
768     return club_name;
769 }
770
771 //best club function gets sportman array and a pointer to the number of sportsman
772 //the function prints the club name which has the biggest number of sportsman take part in the events. it also prints the number of events.
773 void bestClub(sportsman* sportsmen_array, int* num_sportsmen)
774 {
775     typedef struct {
776         char Club_Name;
777         int number_of_events;
778     } ClubEvents;
779
780     // array to store club names and their corresponding event counts
781     ClubEvents array_of_clubs_and_their_events[MAX_CLUB_NUMBER];
782     for (int i = 0; i < MAX_CLUB_NUMBER; i++) {
783
784         array_of_clubs_and_their_events[i].Club_Name = NULL;
785         array_of_clubs_and_their_events[i].number_of_events = 0;
786     }
787
788     // running on each sportsman and count their events for each club
789     for (int i = 0; i < *num_sportsmen; i++) {
790
791         sportsman current_sportsman = sportsmen_array[i];
792         char club_name = getSportClubName(current_sportsman.p2club);
793         int found = 0;
794         for (int j = 0; j < MAX_CLUB_NUMBER; j++) {
795
796             if (array_of_clubs_and_their_events[j].Club_Name != NULL &&
797                 strcmp(array_of_clubs_and_their_events[j].Club_Name, club_name) == 0) {
798                 array_of_clubs_and_their_events[j].number_of_events += current_sportsman.Nevents;
799                 found = 1;
800             }
801         }
802         else if (array_of_clubs_and_their_events[j].Club_Name == NULL)
803         {
804             array_of_clubs_and_their_events[j].Club_Name = club_name;
805             array_of_clubs_and_their_events[j].number_of_events += current_sportsman.Nevents;
806             found = 1;
807         }
808     }
809
810     if (!found) {
811         free(club_name); // free allocated memory
812     }
813 }
814
815 // find the club with the maximum number of events
816 char* best_club_name = NULL;
817 int max_events = 0;
818 for (int i = 0; i < MAX_CLUB_NUMBER; i++)
819 {
820     if (array_of_clubs_and_their_events[i].Club_Name != NULL &&
821         array_of_clubs_and_their_events[i].number_of_events > max_events) {
822         best_club_name = array_of_clubs_and_their_events[i].Club_Name;
823         max_events = array_of_clubs_and_their_events[i].number_of_events;
824     }
825 }
826 printf("The club which has the biggest number of sportsmen who participated in the events is %s and the number of events is: %d\n", best_club_name, max_events);
827

```

```

828 //help function to correct the forman of last name
829 void correctFormatLastName(char* lastName)
830 {
831     int i;
832
833     // Convert all letters to lowercase
834     for (i = 0; lastName[i] != '\0'; i++) {
835         lastName[i] = tolower(lastName[i]);
836     }
837
838     // Capitalize the first letter
839     if (i > 0) {
840         lastName[0] = toupper(lastName[0]);
841     }
842 }
843
844 //help function to capitalize the FirstLetter of given word
845 void capitalizeFirstLetter(char* str) {
846     int i;
847
848     // capitalize first letter of the string
849     if (strlen(str) > 0) {
850         str[0] = toupper(str[0]);
851     }
852
853     // capitalize first letter of each word
854     for (i = 1; str[i] != '\0'; i++) {
855         if ((isspace(str[i - 1]) && isalpha(str[i]))) {
856             str[i] = toupper(str[i]);
857         }
858         else {
859             str[i] = tolower(str[i]);
860         }
861     }
862 }
863
864 //CheckSameEvent function gets id number (I), sportsmen array and number of sportsman
865 //the function check if other sportsmens participated in the same events with this sportman.
866 //if there is other sportsmen that participated in the same events with this sportman the function print the event and the names and return 1/0
867 int CheckSameEvent(int I, sportsman* sportsmen_array, int* num_sportsmen) {
868     int flag = 0;
869     sportsman* current_sportsman = NULL;
870
871     event current_event;
872     for (int i = 0; i < *num_sportsmen; i++) {
873         if (sportsmen_array[i].id == I) {
874             current_sportsman = &sportsmen_array[i];
875             flag = 1;
876             break;
877         }
878     }
879     // if sportsman with given ID is not found, print message and return 0
880     if (current_sportsman == NULL) {
881         printf("Sportsman with ID %d not found.\n", I);
882         return 0;
883     }
884     if (strcmp(current_sportsman->p2events, "No events registered") == 0) {
885         printf("Sportsman with ID %d has no events.\n", I);
886         return 0;
887     }
888     // running on the events of the current sportsman
889     for (int i = 0; i < current_sportsman->Nevents; i++)
890     {
891         current_event = current_sportsman->p2events[i];
892
893         int participated_with_others = 0; // flag that checks if the current sportsman participated with other sportsmen in this event
894
895         // running on other sportsmen to find those who participated in the same event
896         for (int j = 0; j < *num_sportsmen; j++)
897         {
898             if (sportsmen_array[j].id == I)
899             {
900                 continue; // Skip the current sportsman
901
902                 // running on the events of the other sportsman
903                 for (int k = 0; k < sportsmen_array[j].Nevents; k++)
904                 {
905                     event other_event = sportsmen_array[j].p2events[k];
906                     if (strcmp(current_event.p2title, other_event.p2title) == 0 &&
907                         strcmp(current_event.p2location, other_event.p2location) == 0 &&
908                         current_event.year == other_event.year)
909                     {
910                         // print the first name and last name of the other sportsman who participated in the same event
911                         printf("Event: %s, Location: %s, Year: %d\n", current_event.p2title, current_event.p2location, current_event.year);
912                         printf("Participant: %s %s", sportsmen_array[j].Fname, sportsmen_array[j].Lname);
913                         flag = 1; // Set flag to indicate success
914                         participated_with_others = 1;
915                     }
916                 }
917             }
918             // if the current sportsman participated with others in this event, print a separator line
919             if (participated_with_others) {
920                 printf("-----\n");
921             }
922         }
923     }
924     // if flag == 0, no other sportsmen participated in events with the specified sportsman
925     if (flag == 0)
926     {
927         printf("No other sportsmen participated in events with sportsman ID %d.\n", I);
928     }
929     return flag; // return flag indicating success or failure (0/1)
930 }
931
932 // function to print sorted unique events for a given club
933 void printSortedEvents(const char* C, sportsman* sportsmen_array, int* num_sportsmen) {
934     // flag to check if the club exist or not
935     int club_exists = 0;
936
937     // array to store unique events for the club
938     event club_events_arr[MAX_EVENT_NUMBER];
939     int club_number_of_events = 0;
940
941     // running on the sportsmen_array
942     for (int i = 0; i < *num_sportsmen; i++) {
943         // Check if the current sportsman belongs to the club
944         if (strcmp(sportsmen_array[i].p2club, C) == 0) {
945             club_exists = 1; // Club exists
946
947             // running on the events array of the current sportsman
948             for (int j = 0; j < sportsmen_array[i].Nevents; j++) {
949                 // check if the current event is already in the set of events of this club
950                 int event_exists = 0;
951                 for (int k = 0; k < club_number_of_events; k++) {
952                     if (strcmp(club_events_arr[k].p2title, sportsmen_array[i].p2events[j].p2title) == 0 &&
953                         strcmp(club_events_arr[k].p2location, sportsmen_array[i].p2events[j].p2location) == 0 &&
954                         club_events_arr[k].year == sportsmen_array[i].p2events[j].year) {
955                             event_exists = 1; // event already exists
956                             break;
957                         }
958
959                     // if the event is not already in the array, add it
960                     if (!event_exists) {
961                         club_events_arr[club_number_of_events] = sportsmen_array[i].p2events[j];
962                         club_number_of_events++;
963                     }
964                 }
965             }
966         }
967     }

```

```

963     }
964 
965     // If the club does not exist, print a message to the user
966     if (!club_exists)
967     {
968         printf("Club '%s' does not exist.\n", C);
969         return;
970     }
971 
972     // a bubble sort to sort the events by year (we know bubble sort from high school, so we used it)
973     for (int x = 0; x < club_number_of_events - 1; x++)
974     {
975         for (int y = 0; y < club_number_of_events - x - 1; y++)
976         {
977             if (club_events_arr[y].year > club_events_arr[y + 1].year) {
978                 // swap events if the current is bigger than the next
979                 event temp = club_events_arr[y];
980                 club_events_arr[y] = club_events_arr[y + 1];
981                 club_events_arr[y + 1] = temp;
982             }
983         }
984     }
985 
986     // print the sorted array
987     for (int m = 0; m < club_number_of_events; m++)
988     {
989         printf("Event: %s, Year: %d\n", club_events_arr[m].p2title, club_events_arr[m].year);
990     }
991 }

992 void deleteEvent(const char* E, int Y, sportsmen* sportsmen_array, int* num_sportsmen) {
993 
994     int is_exist = 0; //flag
995     for (int i = 0; i < *num_sportsmen; i++) {
996         for (int j = 0; j < sportsmen_array[i].Nevents; j++)
997         {
998             // check if the event matches the given name (E) and year (Y)
999             if (strcmp(sportsmen_array[i].p2events[j].p2title, E) == 0 && sportsmen_array[i].p2events[j].year == Y) {
1000                 is_exist = 1;
1001                 free(sportsmen_array[i].p2events[j].p2title);
1002                 free(sportsmen_array[i].p2events[j].p2location);
1003 
1004                 // remove the event from the events list
1005                 for (int k = j; k < sportsmen_array[i].Nevents - 1; k++)
1006                 {
1007                     sportsmen_array[i].p2events[k] = sportsmen_array[i].p2events[k + 1];
1008                 }
1009                 sportsmen_array[i].Nevents--; // decrement the number of events
1010                 sportsmen_array[i].p2events = realloc(sportsmen_array[i].p2events, sportsmen_array[i].Nevents * sizeof(event));
1011                 j--; // -1 for nevents
1012             }
1013         }
1014     }
1015     if (is_exist)
1016     {
1017         printf("Event '%s' in year %d deleted successfully.\n", E, Y);
1018     }
1019     else
1020     {
1021         printf("Event '%s' in year %d does not exist.\n", E, Y);
1022     }
1023 }

1024 //NewClub Function gets club1 name, club2 name, sportsmen_array and a pointer number of sportsmen
1025 //the function create file new file "Club.txt" which contains the events of sportsman from club1 and club2
1026 void NewClub(const char* C1, const char* C2, sportsmen* sportsmen_array, int* num_sportsmen)
1027 {
1028 
1029     FILE* fp = fopen("Club.txt", "w"); //create new file with the name "Club.txt"
1030     if (fp == NULL)
1031     {
1032         printf("Error opening file.\n");
1033         return;
1034     }
1035 
1036     event eventsC1[MAX_EVENT_NUMBER]; //events array for C1
1037     int numEventsC1 = 0;
1038     event eventsC2[MAX_EVENT_NUMBER]; //events array for C2
1039     int numEventsC2 = 0;
1040 
1041     // running on sportsmen_array
1042     for (int i = 0; i < *num_sportsmen; i++)
1043     {
1044         if (strcmp(sportsmen_array[i].p2club, C1) == 0)
1045         {
1046 
1047             // check if the event already exists in eventsC1
1048             int eventExists = 0; //Flag
1049             for (int k = 0; k < numEventsC1; k++)
1050             {
1051                 if (strcmp(eventsC1[k].p2title, sportsmen_array[i].p2events[j].p2title) == 0 &&
1052                     strcmp(eventsC1[k].p2location, sportsmen_array[i].p2events[j].p2location) == 0 &&
1053                     eventsC1[k].year == sportsmen_array[i].p2events[j].year) {
1054                     eventExists = 1; //event already exists in eventsC1 array
1055                     break;
1056                 }
1057             }
1058             // if the event doesn't exist, add it to eventsC1
1059             if (!eventExists)
1060             {
1061                 eventsC1[numEventsC1] = sportsmen_array[i].p2events[j];
1062                 numEventsC1++;
1063                 fprintf(fp, "%s,%s,%d\n", sportsmen_array[i].p2events[j].p2title, sportsmen_array[i].p2events[j].p2location, sportsmen_array[i].p2events[j].year);
1064             }
1065 
1066         }
1067     }
1068 
1069     else if (strcmp(sportsmen_array[i].p2club, C2) == 0)
1070     {
1071         for (int j = 0; j < sportsmen_array[i].Nevents; j++)
1072         {
1073             // Check if the event already exists in eventsC2
1074             int eventExists = 0; //Flag
1075             for (int k = 0; k < numEventsC2; k++)
1076             {
1077                 if (strcmp(eventsC2[k].p2title, sportsmen_array[i].p2events[j].p2title) == 0 &&
1078                     strcmp(eventsC2[k].p2location, sportsmen_array[i].p2events[j].p2location) == 0 &&
1079                     eventsC2[k].year == sportsmen_array[i].p2events[j].year) {
1080                     eventExists = 1; //event already exists
1081                     break;
1082                 }
1083             }
1084         }
1085     }
1086 }
```

```

1080         eventExists = 1; //event already exists
1081         break;
1082     }
1083     // if the event doesn't exist, add it to eventsC2
1084     if (!eventExists)
1085     {
1086         eventsC2[numEventsC2] = sportsmen_array[i].p2events[j];
1087         numEventsC2++;
1088         fprintf(fp, "%s,%s,%d\n", sportsmen_array[i].p2events[j].p2title, sportsmen_array[i].p2events[j].p2location, sportsmen_array[i].p2events[j].year);
1089     }
1090 }
1091 }
1092 }
1093 }
1094
1095 fclose(fp); // close the file
1096 printf("Club.txt created successfully.\n");
1097 }
1098
1099
1099 //print menu function
1100 int printMenu()
1101 {
1102     printf("(0) Exit the program\n");
1103     printf("(1) Add a new sportsman\n");
1104     printf("(2) Add a new event\n");
1105     printf("(3) Print sportsman array\n");
1106     printf("(4) Print for each sportsman his participated events\n");
1107     printf("(5) Print specific sportsman events\n");
1108     printf("(6) Print the number of sportsmen");
1109     printf("(7) Print the year\n");
1110     printf("(8) Check the same events");
1111     printf("(9) Print sorted events");
1112     printf("(10) Delete an event");
1113     printf("(11) Activate new club function\n\n");
1114     printf("Enter your choice:");
1115 }
1116
1117
1117 //free the sportsmen_array and the event_array for each sportmen
1118 void freeSportsmenArray(Sportsman* sportsmen_array, int* num_sportsmen)
1119 {
1120     //running on sportsmen array
1121     for (int i = 0; i < *(num_sportsmen); i++)
1122     {
1123         for (int j = 0; j < sportsmen_array[i].nevents; j++) //running on events array for each sportsman
1124         {
1125             free(sportsmen_array[i].p2events[j].p2title); //free the event title (cause its dynamic title)
1126             free(sportsmen_array[i].p2events[j].p2location); //free the event location (cause its dynamic location)
1127         }
1128         free(sportsmen_array[i].p2events); //free all the events array for each sports man
1129     }
1130     free(sportsmen_array); //after that free the array of sports man.
1131 }
1132
1133
1133 void main()
1134 {
1135     int num_sportsmen = countLines("SportsmanData.txt"); //number of sportsmen - taken from SportsmanData file.
1136     Sportsman* sportsmen_array = NULL; //pointer to sportsmen array
1137     FromFile2Sportsman("SportsmanData.txt", &sportsmen_array, num_sportsmen); //get the sportsman array
1138     FromFile2Events("EventData.txt", &sportsmen_array, num_sportsmen); //insert array of events for each sportsman
1139     printf("-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----\n");
1140     int choice;
1141     int selected_id;
1142     while (1)
1143     {
1144         printMenu();
1145         scanf("%d", &choice);
1146         while (getchar() != '\n');
1147         switch (choice)
1148         {
1149             case 0:
1150                 printf("Exiting!\n");
1151                 exit(0);
1152             case 1:
1153                 if (addSportsman(&sportsmen_array, &num_sportsmen) == 1)
1154                 {
1155                     printf("Sportsman added successfully.\n");
1156                     Write_Array_to_Sportsman_data("SportsmanData.txt", sportsmen_array, num_sportsmen);
1157                 }
1158                 else
1159                 {
1160                     printf("Failed to add sportsman.\n");
1161                 }
1162                 break;
1163             case 2:
1164                 while (1)
1165                 {
1166                     printf("Enter the sportsman ID for whom you want to add an event: ");
1167                     if (scanf("%d", &selected_id) != 1) {
1168                         printf("Error: Invalid input for ID. Please enter a number.\n");
1169                         while (getchar() != '\n');
1170                     }
1171                     else {
1172                         break;
1173                     }
1174                 }
1175                 if (addEvent(&sportsmen_array, selected_id, &num_sportsmen) == 1) |
1176                 {
1177                     printf("Event added successfully.\n");
1178                     Write_Array_to_Event_Data("EventData.txt", sportsmen_array, num_sportsmen);
1179                 }
1180                 else {
1181                     printf("Failed to add event.\n");
1182                 }
1183                 break;
1184             case 3:
1185                 printSportsmen(sportsmen_array, &num_sportsmen);
1186                 break;
1187             case 4:
1188                 print_Sportman_participated_events(sportsmen_array, &num_sportsmen);
1189                 break;
1190         }
1191     }
1192 }

```

```

1182     case 5:
1183         printf("Enter the last name of the sportsman:");
1184         char sportsman_last_name[MAX_NAME_LENGTH];
1185         scanf("%s", sportsman_last_name);
1186         if (atoi(sportsman_last_name) != 0) {
1187             printf("Invalid input: Last name cannot be an integer.\n");
1188             return 1;
1189         }
1190         if (does_string_has_number(sportsman_last_name) == 1) {
1191             printf("Invalid input: Last name cannot be an integer.\n");
1192             return 1;
1193         }
1194         capitalize_each_First_Letter(sportsman_last_name);
1195         clearInputBuffer(sportsman_last_name);
1196         int result = printEvents(sportsman_last_name, sportsmen_array, &num_sportsmen);
1197         if (result == 0)
1198         {
1199             printf("Events list is empty for this sportsman.\n");
1200         }
1201         else if (result == -1)
1202         {
1203             printf("There is no sportsman with this last name\n");
1204         }
1205         else
1206         {
1207             printf("Events printed successfully\n");
1208         }
1209     break;
1210 }

1211     case 6:
1212     {
1213         printf("Enter the specific event name:");
1214         char specific_event_name[MAX_NAME_LENGTH];
1215         fgets(specific_event_name, sizeof(specific_event_name), stdin);
1216         capitalizeFirstLetter(specific_event_name);
1217         specific_event_name[strlen(specific_event_name, "\n")] = '\0';
1218         int specific_event_year;
1219         printf("Enter the specific event year:");
1220         scanf("%d", &specific_event_year);
1221         while (getchar() != '\n');
1222         printf("The number of sportsmen who participated in %s in %d is: %d\n", specific_event_name, specific_event_year, countEvent(specific_event_name, specific_event_year));
1223     break;
1224 }
1225     case 7:
1226     {
1227         bestClub(sportsmen_array, &num_sportsmen);
1228     break;
1229 }
1230     case 8:
1231     {
1232         int I;
1233         printf("Enter the sportsman ID for whom you want to check: ");
1234         scanf("%d", &I);
1235         printf("Here is the list of sportsmen that participated in the same event with this sportsman:\n");
1236         CheckSameEvent(I, sportsmen_array, &num_sportsmen);
1237     break;
1238 }

1239     case 9:
1240     {
1241         printf("Enter the club name:");
1242         char club_name[MAX_NAME_LENGTH];
1243         fgets(club_name, sizeof(club_name), stdin);
1244         capitalize_each_First_Letter(club_name);
1245         club_name[strlen(club_name, "\n")] = '\0';
1246         printSortedEvents(club_name, sportsmen_array, &num_sportsmen);
1247     break;
1248 }
1249     case 10:
1250     {
1251         printf("Enter the event name that you want to delete:");
1252         char E[MAX_NAME_LENGTH];
1253         fgets(E, sizeof(E), stdin);
1254         capitalize_each_First_Letter(E);
1255         E[strlen(E, "\n")] = '\0';
1256         int Y;
1257         printf("Enter the event year that you want to delete:");
1258         scanf("%d", &Y);
1259         while (getchar() != '\n');
1260         deleteEvent(E, Y, sportsmen_array, &num_sportsmen);
1261         Write_array_to_Event_Data("EventData.txt", sportsmen_array, num_sportsmen);
1262     break;
1263 }

1264     case 11:
1265     {
1266         printf("Enter the first club name:");
1267         char C1[MAX_NAME_LENGTH];
1268         fgets(C1, sizeof(C1), stdin);
1269         capitalize_each_First_Letter(C1);
1270         C1[strlen(C1, "\n")] = '\0';
1271         printf("Enter the second club name:");
1272         char C2[MAX_NAME_LENGTH];
1273         fgets(C2, sizeof(C2), stdin);
1274         capitalize_each_First_Letter(C2);
1275         C2[strlen(C2, "\n")] = '\0';
1276         NewClub(C1, C2, sportsmen_array, &num_sportsmen);
1277     break;
1278 }
1279     default:
1280     {
1281         printf("Invalid choice. Please enter a number between 0 and 11.\n");
1282     break;
1283 }
1284 }

1285 freeSportsmenArray(sportsmen_array, &num_sportsmen);
1286 }
```

צלומי מסך של פלטיפ:

קובץ הספורטאים CSV שקיבלו אותו:

```
SportsmanData.txt - Notepad
File Edit Format View Help
formatsportsman,id;first_name;last_name;club_name;gender
1;Rina;Zviel;Maccabi Haifa;1
12;Benjamin;Salomon;Betar Jerusalem;0
3;Avraham;Shutub;Hapoel Haifa;0
5;Arieli;Shamir;Maccabi Hod Hasharon;0
6;Efrat;Neter;Maccabi Haifa;1
4;Oziza;Beimel;Maccabi Hod Hasharon;1
21;Chen;Shaham;Hapoel Haifa;1
2;Dana;Metzger;Maccabi Netanya;1
7;Arie;Reichman;Hapoel Tel Aviv;0
8;Dafna;Schwartz;Hapoel Haifa;1
9;Michal;Armoni;Hapoel Haifa;1
10;Abaron;Gero;Maccabi Haifa;0
18;Gidi;Cohen;Maccabi Netanya;0
19;Inna;Gefen;Maccabi Hod Hasharon;1
23;Iotam;Evron;Hapoel Tel Aviv;0
```

קובץ אירועים ספורט CSV שקיבלו אותו:

```
EventData.txt - Notepad
File Edit Format View Help
College Championship,France,2020
1,World Cup,Belgium,2011
1,World Championship,Portugal,2018
1,European Championship,UK,2011
1,National Championship,Israel,2013
1,Moshe Rabin-Haniel,2023
12,Bbs,Ccc,1949
3,World Cup,USA,2006
3,European Championship,Germany,2019
3,European Cup,Belgium,2011
3,National Championship,Israel,2009
3,World Cup,Portugal,2007
5,European Cup,Portugal,2008
5,European Cup,Portugal,2010
5,European Cup,France,2010
6,European Championship,Canada,2012
6,World Cup,USA,2004
6,European Championship,UK,2016
6,European Cup,Portugal,2006
6,College Championship,USA,2017
4,National Championship,Israel,2020
4,European Championship,Germany,2019
4,World Cup,Spain,2014
2,College Championship,Belgium,2011
2,National Championship,Israel,2020
21,College Championship,France,2020
21,World Cup,UK,2011
2,European Cup,Belgium,2013
2,European Cup,Portugal,Israel,2015
2,World Cup,UK,2014
7,World Championship,Canada,2012
7,World Cup,Spain,2014
7,European Cup,Portugal,2011
9,College Championship,USA,2016
9,College Championship,USA,2012
10,College Championship,UK,2017
10,European Championship,UK,2016
10,College Championship,Canada,2012
10,European Championship,Belgium,2011
18,European Championship,Germany,2019
18,Euro,UK,2020
19,National Championship,Israel,2015
19,European Cup,Portugal,2008
19,European Cup,Portugal,2010
19,World Cup,Spain,2014
23,World Cup,USA,2006
```

הרצאה לדוגמה בlijivo מסכי פלטיפ

cutt ננסה להויסיף ספורטאי חדש לקובץ הספורטאים:

```
C:\Users\user\Desktop\Assignment\Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:

(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:1
Please enter the new sportsman details:
Enter the new sportsman ID: 99
Enter the new sportsman First name: niv
Enter the new sportsman Last name: alex
Enter the new sportsman Club name: maccabi ruppin
Enter the new sportsman Gender (1 for male or 0 for female): 1
Sportsman added successfully.
Array successfully written to SportsmanData.txt.
Please choose an option:
```

הויספנו את הספורטאי הבא לקובץ הספורטאים:

תעודת זהות: 99

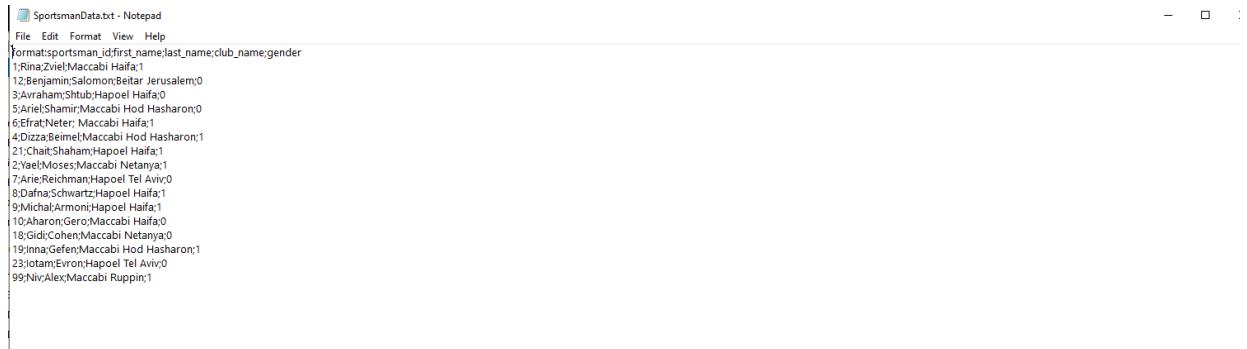
שם פרטי: ניב

שם משפחה: alex

מועדון ספורט: Maccabi ruppin

מין: 1 (זכר)

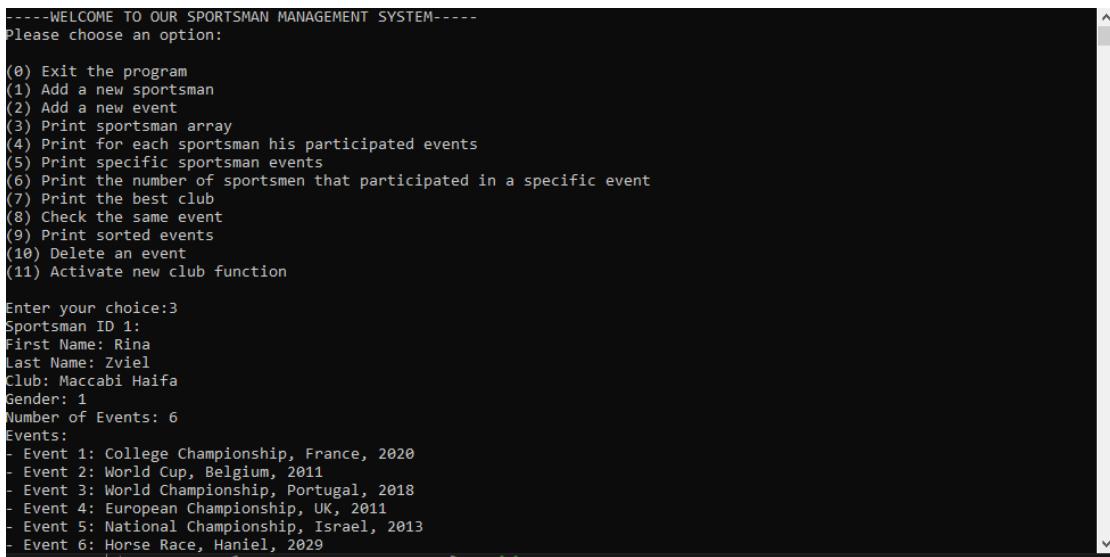
כעת נציג את קובץ הספורטאים המעודכן:



```
SportsmanData.txt - Notepad
File Edit Format View Help
format:sportsman_id;first_name;last_name;club_name;gender
1;Rina;Zviel|Maccabi Haifa;1
12;Benjamin;Salomon|Beitar Jerusalem;0
3;Avraham;Shtub|Hapoel Haifa;0
5;Ariel;Shamir|Maccabi Hod Hasharon;0
6;Efrat;Neter; Maccabi Haifa;1
4;Dizza;Beimel|Maccabi Hod Hasharon;1
21;Chit;Shaham|Hapoel Haifa;1
2;Yael;Moses|Maccabi Netanya;1
7;Ari;Reichman|Hapoel Tel Aviv;0
8;Dafna;Schwartz|Hapoel Haifa;1
9;Michal;Armoni|Hapoel Haifa;1
10;Aharon;Gero|Maccabi Haifa;0
18;Gidi;Cohen|Maccabi Netanya;0
19;Inna;Gefen|Maccabi Hod Hasharon;0
23;Iotam;Evron|Hapoel Tel Aviv;0
99;Niv;Alex|Maccabi Ruppin;1
```

ניתן לראות כי בשורה האחרונה אכן התווסף ספורטאי בשם ניב אלכס, עם
תעודת זהות 99 וקובצת ספורט של מכבי רופין (שים לב שהכנסו לקובץ
בפורמט תקין, וכל שם מתחילה באות גודלה)

כעת נציג בתוכנית את מערך הספורטאים המעודכן:



```
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:

(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:3
Sportsman ID 1:
First Name: Rina
Last Name: Zviel
Club: Maccabi Haifa
Gender: 1
Number of Events: 6
Events:
- Event 1: College Championship, France, 2020
- Event 2: World Cup, Belgium, 2011
- Event 3: World Championship, Portugal, 2018
- Event 4: European Championship, UK, 2011
- Event 5: National Championship, Israel, 2013
- Event 6: Horse Race, Haniel, 2029
```

```
C:\Users\user\העבורה\שולחן\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe

Sportsman ID 12:
First Name: Benjamin
Last Name: Salomon
Club: Beitar Jerusalem
Gender: 0
Number of Events: 1
Events:
- Event 1: Bbb, Ccc, 1949

Sportsman ID 3:
First Name: Avraham
Last Name: Shtub
Club: Hapoel Haifa
Gender: 0
Number of Events: 5
Events:
- Event 1: World Cup, USA, 2006
- Event 2: European Championship, Germany, 2019
- Event 3: European Cup, Belgium, 2011
- Event 4: National Championship, Israel, 2009
- Event 5: World Cup, Portugal, 2017

Sportsman ID 5:
First Name: Ariel
Last Name: Shamir
Club: Maccabi Hod Hasharon
Gender: 0
Number of Events: 3
Events:
```

```
C:\Users\user\העבורה\שולחן\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe

- Event 1: European Cup, Portugal, 2008
- Event 2: Euro, UK, 2020
- Event 3: World Cup, France, 2010

Sportsman ID 6:
First Name: Efrat
Last Name: Neter
Club: Maccabi Haifa
Gender: 1
Number of Events: 5
Events:
- Event 1: European Championship, Canada, 2012
- Event 2: World Cup, USA, 2006
- Event 3: European Championship, UK, 2016
- Event 4: European Cup, Portugal, 2008
- Event 5: College National Championship, USA, 2017

Sportsman ID 4:
First Name: Dizza
Last Name: Beimel
Club: Maccabi Hod Hasharon
Gender: 1
Number of Events: 3
Events:
- Event 1: National Championship, Israel, 2020
- Event 2: European Championship, Germany, 2019
- Event 3: World Cup, Spain, 2014
```

```
C:\Users\user\העבורה\שולחן\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe

Sportsman ID 21:
First Name: Chait
Last Name: Shaham
Club: Hapoel Haifa
Gender: 1
Number of Events: 4
Events:
- Event 1: European Championship, Belgium, 2011
- Event 2: National Championship, Israel, 2020
- Event 3: College Championship, France, 2020
- Event 4: World Cup, UK, 2014

Sportsman ID 2:
First Name: Yael
Last Name: Moses
Club: Maccabi Netanya
Gender: 1
Number of Events: 3
Events:
- Event 1: European Cup, Belgium, 2013
- Event 2: National Championship, Israel, 2015
- Event 3: World Cup, UK, 2014

Sportsman ID 7:
First Name: Arie
Last Name: Reichman
Club: Hapoel Tel Aviv
Gender: 0
Number of Events: 3
```

```
Sportsman ID 8:  
First Name: Dafna  
Last Name: Schwartz  
Club: Hapoel Haifa  
Gender: 1  
Number of Events: 0  
Events:  
No events registered  
  
Sportsman ID 9:  
First Name: Michal  
Last Name: Armoni  
Club: Hapoel Haifa  
Gender: 1  
Number of Events: 2  
Events:  
- Event 1: World Championship, USA, 2016  
- Event 2: College Championship, Canada, 2012
```

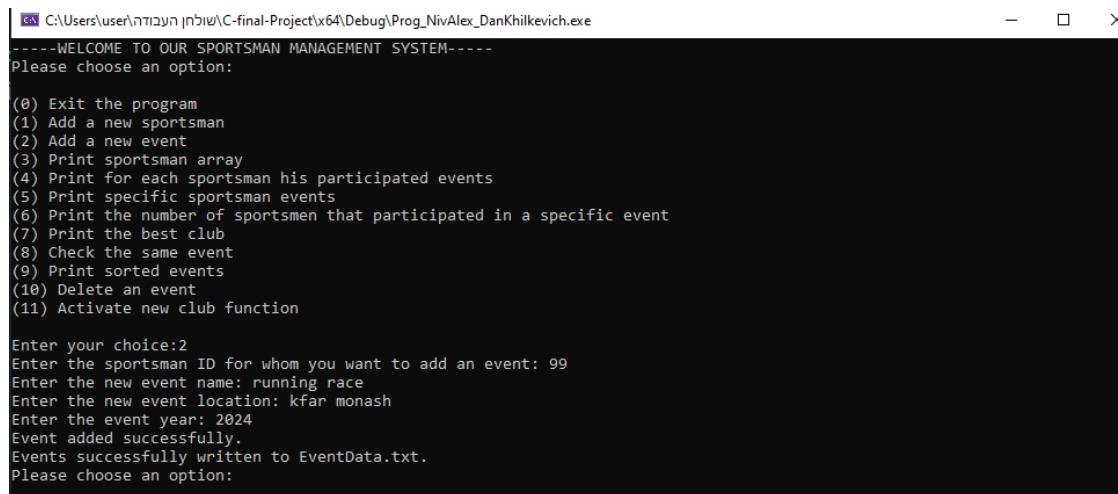
```
1 C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe  
Sportsman ID 10:  
First Name: Aharon  
Last Name: Gero  
Club: Maccabi Haifa  
Gender: 0  
Number of Events: 3  
Events:  
- Event 1: College Championship, UK, 2017  
- Event 2: European Championship, UK, 2016  
- Event 3: College Championship, Canada, 2012  
  
Sportsman ID 18:  
First Name: Gidi  
Last Name: Cohen  
Club: Maccabi Netanya  
Gender: 0  
Number of Events: 3  
Events:  
- Event 1: European Championship, Belgium, 2011  
- Event 2: European Championship, Germany, 2019  
- Event 3: Euro, UK, 2020  
  
Sportsman ID 19:  
First Name: Inna  
Last Name: Gefen  
Club: Maccabi Hod Hasharon  
Gender: 1  
Number of Events: 4  
Events:  
- Event 1: National Championship, Israel, 2015
```

```
Sportsman ID 23:  
First Name: Iotam  
Last Name: Evron  
Club: Hapoel Tel Aviv  
Gender: 0  
Number of Events: 1  
Events:  
- Event 1: World Cup, USA, 2006  
  
Sportsman ID 99:  
First Name: Niv  
Last Name: Alex  
Club: Maccabi Ruppin  
Gender: 1  
Number of Events: 0  
Events:  
No events registered  
  
Please choose an option:  
(0) Exit the program  
(1) Add a new sportsman  
(2) Add a new event  
(3) Print sportsman array  
(4) Print for each sportsman his participated events  
(5) Print specific sportsman events  
(6) Print the number of sportsmen that participated in a specific event  
(7) Print the best club  
(8) Check the same event
```

שיםו לב כי גם דרך שורת הפקודות כאשר נבחר 3 יודפס המערך המעודכן ואכן אפשר לראות גם דרך שורת הפקודות שנוסיף ספורטאי למערך הספורטאים. השם של הספורטאי הוא Alex Niv. הקבוצה שלו היא Maccabi Ruppin. המין שלו הוא 1, וכרגע יש לו 0 אירועים בהם הוא משתתף.

- נציג כי בעת הכנסת ספורטאי הקפדנו על בדיקת קלט, כך שאם תנוט להכנס ספורטאי עם תעודה זהות "aaa" או עם מגדר "5" לא תצליחו לעשות זאת

כעת ננסה להוסיף אירוע חדש לספורטאי Alex Niv. בנוסף לו מרווח ריצה בכפר מונש בשנת 2024 :

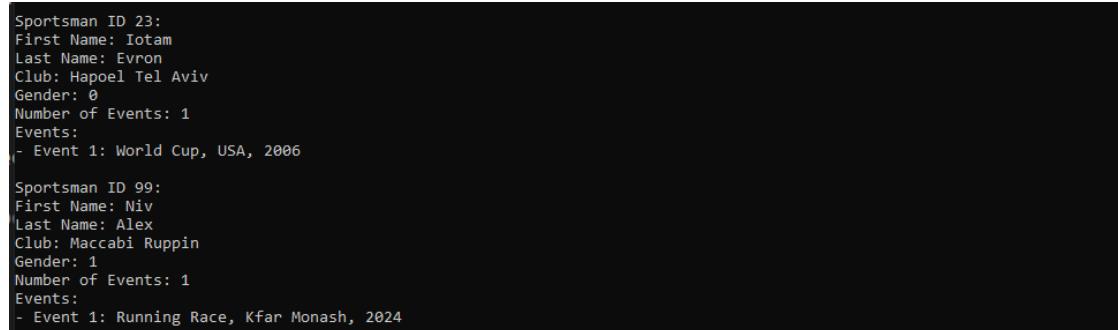


```
C:\Users\user\Desktop\שולחן העבודה\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:

(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:2
Enter the sportsman ID for whom you want to add an event: 99
Enter the new event name: running race
Enter the new event location: kfar monash
Enter the event year: 2024
Event added successfully.
Events successfully written to EventData.txt.
Please choose an option:
```

כעת נדפיס שוב את מידע הספורטאים בשורת הפקודות ונראה כי נוסף אירוע חדש לספורטאי Alex Niv (למען הנוחות הצלום מסך יכלול רק את החלק של הספורטאי הרלוונטי)



```
Sportsman ID 23:
First Name: Iotam
Last Name: Evron
Club: Hapoel Tel Aviv
Gender: 0
Number of Events: 1
Events:
- Event 1: World Cup, USA, 2006

Sportsman ID 99:
First Name: Niv
Last Name: Alex
Club: Maccabi Ruppin
Gender: 1
Number of Events: 1
Events:
- Event 1: Running Race, Kfar Monash, 2024
```

נשים לב כי אכן נוסף לספורטאי Alex Niv אירוע חדש. כעת נראה אם אירוע זה התעדכן בקובץ האירועים.

EventData.txt - Notepad

```

File Edit Format View Help
12.Bbb,Ccc,1946
3.World Cup,USA,2006
3.European Championship,Germany,2019
3.European Cup,Belgium,2011
3.National Championship,Israel,2009
3.World Cup,Portugal,2017
5.European Cup,Portugal,2008
5.Euro,UK,2020
5.World Cup,France,2010
6.European Championship,Canada,2012
6.World Cup,USA,2006
6.European Championship,UK,2016
6.European Cup,Portugal,2006
6.College National Championship,USA,2017
4.National Championship,Israel,2020
4.European Championship,Germany,2019
4.World Cup,Spain,2014
21.European Championship,Belgium,2011
21.National Championship,Israel,2020
21.College Championship,France,2020
21.World Cup,UK,2014
21.Club National,Belgium,2013
2.National Championship,Israel,2015
2.World Cup,UK,2014
7.World Championship,Canada,2012
7.World Cup,Spain,2014
7.European Cup,Portugal,2011
9.World Championship,USA,2016
9.College Championship,Canada,2012
10.College Championship,UK,2017
10.European Championship,Portugal,2016
10.College Championship,Canada,2012
18.European Championship,Belgium,2011
18.European Championship,Germany,2019
18.Euro,UK,2020
19.National Championship,Israel,2015
19.European Cup,Portugal,2008
19.World Cup,France,2010
19.World Cup,Spain,2014
23.World Cup,USA,2006
99.Running Race,Kfar Monash,2024

```

כפי שניתן לראות, בשורה האחרונה נוסף אירוע חדש. אירוע זה נקרא: Running Race. המיקום שלו הוא: Kfar Monash. השנה שלו היא: 2024 והוא שיר לספורטאי עם תעוזות הזיהות 99 (כאמור ספורטאי זה הוא Niv Alex).

עת נבדוק מה קורה אם רוצים להוסיף שוב את אותו אירוע עבור ניב אלכס, כאמור לפי הגדרת השאלה לעילנו להציג למשתמש הוא כי אירוע זה כבר קיים עבור ניב אלכס.

```

C:\Users\user\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function
)
Enter your choice:2
Enter the sportsman ID for whom you want to add an event: 99
Enter the new event name: running race
Enter the new event location: kfar monash
Enter the event year: 2024
The event already exists for this sportman
)
Failed to add event.
Please choose an option:

```

שים לב שאכן המערכת שלנו יודעת להגיד כי אירוע זה כבר קיים עבור ניב אלכס וכן לא ניתן להוסיף אותו שוב פעם.

למען המשך ההסברים של הפקציית הבאות, נוסיף CUT ספורטאי נוסף בשם
דן חילקייביץ' (ת.ז: 98) מקבוצת הפועל רופין (Hapoel Ruppin) ונוסיף לו 2
אירועים (מרוץ ריצה ברופין בשנת 2024 ומשחק כדורסל בעפולה בשנת
(2026

```
C:\Users\user\Desktop\שולחן העבודה\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:1
Please enter the new sportsman details:
Enter the new sportsman ID: 98
Enter the new sportsman First name: dan
Enter the new sportsman Last name: khilkevich
Enter the new sportsman Club name: hapoel ruppin
Enter the new sportsman Gender (1 for male or 0 for female): 1
Sportsman added successfully.
Array successfully written to SportsmanData.txt.
Please choose an option:
```

```
C:\Users\user\Desktop\שולחן העבודה\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:

(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:2
Enter the sportsman ID for whom you want to add an event: 98
Enter the new event name: running race
Enter the new event location: kfar monash
Enter the event year: 2024
Event added successfully.
Events successfully written to EventData.txt.
Please choose an option:
```

```
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:2
Enter the sportsman ID for whom you want to add an event: 98
Enter the new event name: basketball game
Enter the new event location: afula
Enter the event year: 2023
Event added successfully.
Events successfully written to EventData.txt.
```

cutet נדפס את האירועים של דן חילקיץ' (לפי מה שהראנו מוקדם אמורים להיות לו 2 אירועים: ריצה בכפר מונש ומשחק כדורסל בעפולה). המערכת שלנו תבקש מהמשתמש שם משפחה ותציג את האירועים של הספורטאי.

```
C:\Users\user\Desktop\הובזה\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:5
Enter the last name of the sportsman:khilkevich
Events for Dan Khilkevich:
(-) Running Race, Kfar Monash, 2024
(-) Basketball Game, Afula, 2023
Events printed successfully
Please choose an option:
```

נשים לב כי אכן עברו הקלט: "khilkevich" הוצגו כל אירועי הספורט שבhem דן חילקיץ' משתתף.

cutet נדפס עבור אירוע ספציפי את מספר הספורטאים המשתתפים בו. לדוגמה עבור מרוץ הריצה בכפר מונש אנו מקבלים רק ניב אלכס ודן חילקיץ' המשתתפים בו. נראה כי זה אכן הפלט:

```
C:\Users\user\Desktop\הובזה\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:6
Enter the specific event name:running race
Enter the specific event year:2024
The number of sportsmen who participated in Running Race in 2024 is: 2
Please choose an option:
```

נשים לב כי אכן יש 2 משתתפים באירוע ספורט זה. אם לדוגמה היינו משנים את השנה לשנת 2023, כבר יהיו 0 משתתפים:

```

C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:6
Enter the specific event name:running race
Enter the specific event year:2023
The number of sportsmen who participated in Running Race in 2023 is: 0
Please choose an option:

```

כעת נדפס את המועדון הטוב ביותר. בנתונים הנוכחיים המועדון הטוב ביותר הוא מכבי חיפה (זה הוא אכן המועדון שיש לו את מספר הספורטאים משתמש במספר הרב ביותר של אירועי ספורט)

```

C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:7
The club which has the biggest number of sportsmen who participated in the events is Maccabi Haifa and the number of events is: 14
Please choose an option:

(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events

```

נשים לב שגם נוסיף אירועים לשחקנים מקבוצות ספורט אחרות (דוגמת הפועל ת"א/מכבי ת"א וכו'...) יכול להיות שבשלב מסוים מכבי חיפה כבר לא תהיה הקבוצה הטובה ביותר. לדוגמה לקחנו את מיל ארמוני מהפועל חיפה (תעודת זהות 9) והוספנו לה מלא אירועים. כעת המועדון הטוב ביותר הוא הפועל חיפה.

```

C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:7
The club which has the biggest number of sportsmen who participated in the events is Hapoel Haifa and the number of events is: 22
Please choose an option:

(0) Exit the program

```

כעת עברו ניב אלכס, נבדוק אם קיימ ספורטאי אחר שהשתתף איתו באותו אירע ספורט, כאמור אנחנו מצפים שיהיה ספורטאי זה בשם דן חילקי'ץ (כי שניהם השתתפו במירוץ ריצה בכפר מונש בשנת 2024)

```
C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:8
Enter the sportsman ID for whom you want to check: 99
Here is the list of sportsmen that participated in the same event with this sportsman:
Event: Running Race, Location: Kfar Monash, Year: 2024
Participant: Dan Khilkevich
-----
Please choose an option:
```

ואכן המערכת מדפיסה כי עברו ניב אלכס (ת.ז: 99) יש אירע משותף עם דן חילקי'ץ והוא נקרא מרוץ ריצה בכפר מונש בשנת 2024.

אותו דבר אם נעשה הפור, נבדוק אם קיימ ספורטאי שהשתתף עם דן חילקי'ץ באותו אירע (אנו מצפים לקבל את ניב אלכס).

```
C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:8
Enter the sportsman ID for whom you want to check: 98
Here is the list of sportsmen that participated in the same event with this sportsman:
Event: Running Race, Location: Kfar Monash, Year: 2024
Participant: Niv Alex
-----
Please choose an option:
```

ואכן קיבלנו כי המערכת מדפיסה כי עברו דן חילקי'ץ (ת.ז: 98) יש אירע משותף עם ניב אלכס והוא נקרא מרוץ ריצה בכפר מונש בשנת 2024.

כעת נוסיף מספר אירועים לניב אלכס על מנת להסביר בבהירות את הסעיפים הבאים. האירועים שנוסיף הם (משחק כדורגל בכפר יונה בשנת 1973, משחק טניס בחנייאל בשנת 1990, מרוץ שחיה בננתניה בשנת 2025)

כעת מערך האירועים של ניב אלכס נראה כך:

```
Sportsman ID 99:
First Name: Niv
Last Name: Alex
Club: Maccabi Ruppin
Gender: 1
Number of Events: 4
Events:
- Event 1: Running Race, Kfar Monash, 2024
- Event 2: Football Game, Kfar Yona, 1973
- Event 3: Tennis Game, Haniel, 1990
- Event 4: Swimming Race, Netanya, 2025
```

כזכור, ניב אלכס הוא ספורטאי במועדון מכבי רופין. למען נוחות ההסברים לkrarat ha-sufif ha-ba, נוסיף עוד ספורטאי למכבי רופין. לסתורטאי נקרא see messi ותעודת הזהות שלו תהיה 96. נוסיף לו מס' גם אירוע שנקרא סופר קלאסיון במדריד בשנת 2010.

```
Sportsman ID 96:  
First Name: Leo  
Last Name: Messi  
Club: Maccabi Ruppin  
Gender: 1  
Number of Events: 1  
Events:  
- Event 1: Super Clasico, Madrid, 2010
```

כעת במועדון maccabi ruppin יש 2 ספורטאים: ניב אלכס וליאו מסי.

כעת נפעיל את הפונקציה Print sorted events עבור המועדון Maccabi ruppin. אנו מצפים לקבל את האירועים של כל הספורטאים בקבוצה זו ממויינים בסדר עולה לפי שנה لكن הפלט לו נצפה הוא:

1. משחק כדורגל בכפר יונה בשנת 1973
2. משחק טניס בחניאל בשנת 1990
3. סופר קלאסיון במדריד בשנת 2010
4. מרוץ ריצה בכפר מונש בשנת 2024
5. מרוץ שחיה בנטעניה בשנת 2025

ואכן זה הפלט שקיבלנו עבור maccabi ruppin :

```
C:\Users\user\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----
Please choose an option:
(0) Exit the program
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:9
Enter the club name:maccabi ruppin
Event: Football Game, Year: 1973
Event: Tennis Game, Year: 1990
Event: Super Clasico, Year: 2010
Event: Running Race, Year: 2024
Event: Swimming Race, Year: 2025
Please choose an option:
```

כעת נפעיל את הפונקציה למחיקת אירוע. נמחק את אירוע הסופר קלאסיון של לאו מסי בשנת 2010.

```

C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe
(1) Add a new sportsman
(2) Add a new event
(3) Print sportsman array
(4) Print for each sportsman his participated events
(5) Print specific sportsman events
(6) Print the number of sportsmen that participated in a specific event
(7) Print the best club
(8) Check the same event
(9) Print sorted events
(10) Delete an event
(11) Activate new club function

Enter your choice:10
Enter the event name that you want to delete:super clasico
Enter the event year that you want to delete:2010
Event 'Super Clasico' in year 2010 deleted successfully.
Events successfully written to EventData.txt.
Please choose an option:

```

**כעת נדפיס את מערך הספורטאים ואנו מצפים לראות כי ללאו מסוי לא קיימים
אירועים (האירוע היחיד שהוא לו כבר נמחק)**

```

C:\Users\user\Desktop\C-final-Project\x64\Debug\Prog_NivAlex_DanKhilkevich.exe

Sportsman ID 98:
First Name: Dan
Last Name: Khilkevich
Club: Hapoel Ruppin
Gender: 1
Number of Events: 2
Events:
- Event 1: Running Race, Kfar Monash, 2024
- Event 2: Basketball Game, Afula, 2023

Sportsman ID 96:
First Name: Leo
Last Name: Messi
Club: Maccabi Ruppin
Gender: 1
Number of Events: 0
Events:
No events registered

Please choose an option:

```

ואכן ללאו מסוי אין אירועים. נראה גם כי קובץ האירועים התעדכן בהתאם:

```

EventData.txt - Notepad
File Edit Format View Help
3.World Cup,Portugal,2017
5.European Cup,Portugal,2008
5.Euro,UK,2020
5.Volleyball France,2010
6.European Championship,Canada,2012
6.World Cup USA,2006
6.European Championship UK,2016
6.European Cup Portugal,2008
6.College National Championship,USA,2017
4.National Championship, Israel,2020
4.European Championship,Germany,2019
4.World Cup,Spain,2014
21.National Championship,Belgium,2011
21.National Championship,Israel,2020
21.College Championship,France,2020
21.World Cup,UK,2014
2.European Cup,Belgium,2013
2.National Championship,Israel,2015
2.World Cup,UK,2014
7.World Championship,Canada,2012
7.World Cup,Spain,2014
7.European Cup,Portugal,2011
9.World Championship,USA,2016
9.College Championship,Canada,2012
10.College Championship,UK,2017
10.European Championship,UK,2016
10.College Championship,Canada,2012
18.European Championship,Belgium,2011
18.European Championship,Germany,2019
18.European Championship,France,2014
19.National Championships,Israel,2015
19.European Cup,Portugal,2008
19.World Cup,France,2010
19.World Cup,Spain,2014
23.World Cup,USA,2006
99.Running Race,Kfar Monash,2024
99.Football Game,Kfar Yona,1973
99.Tennis Game,Haniel,1990
99.Swimming Race,Netanya,2025
98.Running Race,Kfar Monash,2024
98.Basketball Game,Afula,2023

```

אכן אירוע שנקרא סופר קלאסיקו במדריד בשנת 2010 נמחק מהקובץ.

cut נפעיל את הפונקציה `newclub`. הפונקציה תקבל שמות של 2 מועדונים ותיצור קובץ חדש בשם "club.txt" בו יהיו רשומים האירועים של הספורטאים מ-2 המועדונים. נפעיל את הפונקציה עבור מכבי רופין והפועל רופין.

הספורטאים של מכבי רופין הם ניב אלכס עם האירועים הבאים:

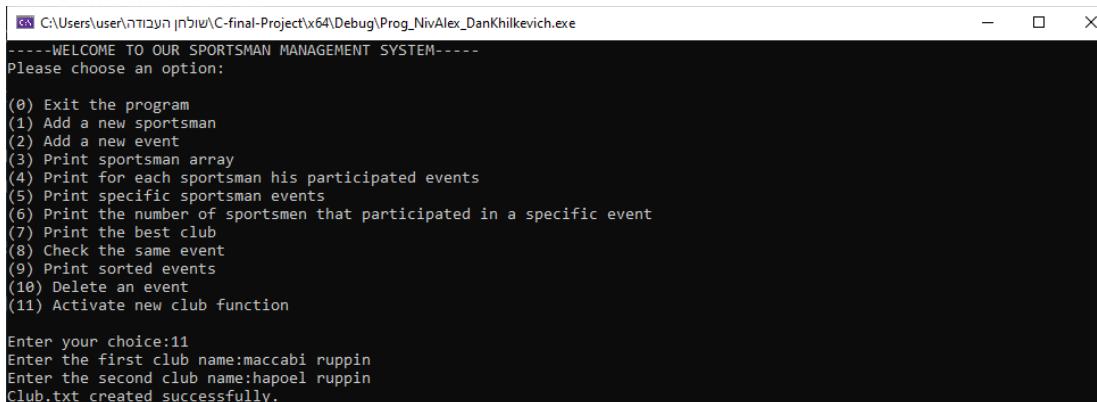
```
Sportsman ID 99:  
First Name: Niv  
Last Name: Alex  
Club: Maccabi Ruppin  
Gender: 1  
Number of Events: 4  
Events:  
- Event 1: Running Race, Kfar Monash, 2024  
- Event 2: Football Game, Kfar Yona, 1973  
- Event 3: Tennis Game, Haniel, 1990  
- Event 4: Swimming Race, Netanya, 2025
```

הספורטאים של הפועל רופין הם דן חילקевич' עם האירועים הבאים:

```
Sportsman ID 98:  
First Name: Dan  
Last Name: Khilkevich  
Club: Hapoel Ruppin  
Gender: 1  
Number of Events: 2  
Events:  
- Event 1: Running Race, Kfar Monash, 2024  
- Event 2: Basketball Game, Afula, 2023
```

נשים לב כי האירוע מרוץ ריצה בכפר מונש מופיע פעמיים (גם אצל ניב וגם אצל דן) אבל בקובץ שיוצג הוא-Amor להופיע פעם אחת בלבד.

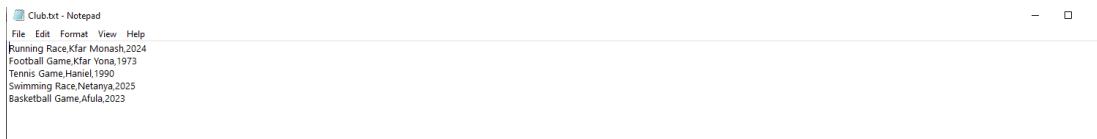
נפעיל את הפונקציה:



```
C:\Users\use\appdata\local\temp\Prog_NivAlex_DanKhilkevich.exe  
-----WELCOME TO OUR SPORTSMAN MANAGEMENT SYSTEM-----  
Please choose an option:  
(0) Exit the program  
(1) Add a new sportsman  
(2) Add a new event  
(3) Print sportsman array  
(4) Print for each sportsman his participated events  
(5) Print specific sportsman events  
(6) Print the number of sportsmen that participated in a specific event  
(7) Print the best club  
(8) Check the same event  
(9) Print sorted events  
(10) Delete an event  
(11) Activate new club function  
  
Enter your choice:11  
Enter the first club name:maccabi ruppin  
Enter the second club name:hapoel ruppin  
Club.txt created successfully.
```

המערכת מודיעה לנו כי הקובץ נוצר בהצלחה.

cut נסתכל על תוכן הקובץ:



```
Club.txt - Notepad  
File Edit Format View Help  
Running Race,Kfar Monash,2024  
Football Game,Kfar Yona,1973  
Tennis Game,Haniel,1990  
Swimming Race,Netanya,2025  
Basketball Game,Afula,2023
```

ואכן האירוע ריצה בכפר מונש בשנת 2024 מודפס רק פעם אחת. בנוסף מודפסים כל שאר האירועים של ניב ודן.

