

# Пројекат из рачунарства

Алекса Вучковић   Леа Ирт   Филип Јеремић

# Задатак

Написати програм који обрађује податке о резултатима фудбалских утамица коришћењем једноструко повезане листе. Податке о резултатима учитати из фајла у ком су подаци представљени на следећи начин, сваки резултат у новој линији: Домаћин, Гост, број голова домаћина, број голова госта (на пример Црвена Звезда, Партизан, 1, 1).

Направити мени за интеракцију са корисником преко конзоле са следећим опцијама који се врти све док корисник не изабере крај рада:

# Задатак

- учитавање података из фајла, име фајла уноси корисник, подаци се учитавају у неуређену листу
- испис свих утакмица из листе
- испис свих утакмица са нерешеним резултатом, исписати утакмице и укупан број
- брисање утакмице госта, корисник уноси назив тима и бришу се све утакмице у којима је он био гост
- испис поена за тим, корисник уноси назив тима, поени се додељују на следећи начин 3-победа, 1-нерешено, 0-пораз
- излаз из програма (обрисату листу из меморије)

## Улазни фајл

Man United,Leicester,2,1  
Bournemouth,Cardiff,2,0  
Fulham,Crystal Palace,0,3  
Huddersfield,Chelsea,0,3  
Newcastle,Tottenham,1,2  
Watford,Brighton,2,0  
Wolves,Everton,2,2  
Arsenal,Man City,0,2  
Liverpool,West Ham,4,0  
Southampton,Burnley,0,0  
Cardiff,Newcastle,0,0  
Chelsea,Arsenal,3,2  
Everton,Southampton,2,1  
Leicester,Wolves,2,0  
Tottenham,Fulham,3,1  
West Ham,Bournemouth,1,2  
Brighton,Man United,3,2  
Burnley,Watford,1,3  
Man City,Huddersfield,6,1  
Crystal Palace,Liverpool,0,2

Arsenal,West Ham,3,1  
Bournemouth,Everton,2,2  
Huddersfield,Cardiff,0,0  
Liverpool,Brighton,1,0  
Southampton,Leicester,1,2  
Wolves,Man City,1,1  
Fulham,Burnley,4,2  
Newcastle,Chelsea,1,2  
Watford,Crystal Palace,2,1  
Man United,Tottenham,0,3  
Brighton,Fulham,2,2  
Chelsea,Bournemouth,2,0  
Crystal Palace,Southampton,0,2  
Everton,Huddersfield,1,1  
Leicester,Liverpool,1,2  
Man City,Newcastle,2,1  
West Ham,Wolves,0,1  
Burnley,Man United,0,2  
Cardiff,Arsenal,2,3  
Partizan,Zvezda,10,0

## Мени

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define MAXN 1000
6
7 struct game {
8     char name1[MAXN];
9     char name2[MAXN];
10    int score1;
11    int score2;
12};
13
14 typedef struct list_item {
15     struct game data;
16     struct list_item *next;
17 } node;
18
19
20 node* novi_cvor(struct game data) {
21     node *novi = malloc(sizeof(node));
22     novi->data = data;
23     novi->next = NULL;
24     return novi;
25 }
26
27 void procitaj_fajl(FILE *f, node **first) {
28     int n = 0;
29     struct game data;
30     char line[MAXN];
31
32     while(fgets(line, sizeof(line), f)) {
33         // procitamo vrednosti pojedinih polja
34         // vrednosti su odvojene zaptama
```

Opције:

- (1) učitavanje podataka iz fajla
- (2) ispis svih utakmica iz liste
- (3) ispis svih utakmica sa neresenim rezultatom
- (4) brisanje utakmica gosta
- (5) ispis poena za tim
- (6) izlaz iz programa

Glavni fajl: in.txt  
Procitano je 39 rezultata utakmica.

Opције:

- (1) učitavanje podataka iz fajla
- (2) ispis svih utakmica iz liste
- (3) ispis svih utakmica sa neresenim rezultatom
- (4) brisanje utakmica gosta
- (5) ispis poena za tim
- (6) izlaz iz programa

Tim: Tottenham  
New United vs Tottenham 0-3  
Tottenham vs Fulham 3-1  
Newcastle vs Tottenham 1-2  
Broj poena: 9

Opције:

- (1) učitavanje podataka iz fajla
- (2) ispis svih utakmica iz liste
- (3) ispis svih utakmica sa neresenim rezultatom
- (4) brisanje utakmica gosta
- (5) ispis poena za tim
- (6) izlaz iz programa

Compiler (10) Resources Compile Log Debug Find Results

Line 1 Col 1 Set 0 Lines: 195 Length: 4650 Insert Done parsing in 0,016 seconds

# Библиотеке и структуре

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXN 1000
struct game {
char name1[MAXN];
char name2[MAXN];
int score1;
int score2;
};
typedef struct list_item {
struct game data;
struct list_item *next;
} node;
```

```
node* novi_cvor(struct game
data) {
node *novi =
malloc(sizeof(node));
novi->data = data;
novi->next = NULL;
return novi;
}
```

# Читање фајла

```
void procitaj_fajl(FILE *f, node **first)
{
    int n = 0;
    struct game data;
    char line[MAXN];
    while(fgets(line, sizeof(line), f)) {
        // procitamo vrednosti pojedinih polja
        // vrednosti su odvojene zapetama
        char *p, *q;
        p = line;
        q = strchr(p, ',');
        *q = '\0';
        strcpy(data.name1, p);
        p = q + 1;
        q = strchr(p, ',');
        *q = '\0';
        strcpy(data.name2, p);
```

```
        p = q + 1;
        q = strchr(p, ',');
        *q = '\0';
        data.score1 = atoi(p);
        p = q + 1;
        data.score2 = atoi(p);
        // napravimo cvor i dodamo na
        pocetak liste
        node *novi = novi_cvor(data);
        novi->next = *first;
        *first = novi;
        n++;
    }
    printf("Procitano je %d rezultata
    utakmica.\n", n);
}
```

## Испис

```
void ispis(node *p) {  
    while(p != NULL) {  
        struct game item = p->data;  
        printf("%s vs %s %d-%d\n",  
            item.name1, item.name2,  
            item.score1, item.score2);  
        p = p->next;  
    }  
}
```

```
void ispis_neresenih(node *p) {  
    while(p != NULL) {  
        struct game item = p->data;  
        if(item.score1 == item.score2)  
            printf("%s vs %s %d-%d\n",  
                item.name1, item.name2,  
                item.score1, item.score2);  
        p = p->next;  
    }  
}
```



## Број поена за тим

```
int broj_poena_za_tim(node *p, char *name) {
    int z = 0;
    while(p != NULL) {
        struct game item = p->data;
        if(strcmp(item.name1, name)==0) {
            printf("%s vs %s %d-%d\n", item.name1, item.name2, item.score1, item.score2);
            if(item.score1 > item.score2)
                z += 3;
            else if(item.score1 == item.score2)
                z += 1;
        }
        else if(strcmp(item.name2, name)==0) {
            printf("%s vs %s %d-%d\n", item.name1, item.name2, item.score1, item.score2);
            if(item.score1 < item.score2)
                z += 3;
            else if(item.score1 == item.score2)
                z += 1;
        }
        p = p->next;
    }
    return z;
}
```

# Брисање листе или њеног дела

```
int brisanje_poena_gosta(node **p, char
*name) {
int z = 0;
while(*p != NULL) {
struct game item = (*p)->data;
if(strcmp(item.name2, name)==0) {
node *t = *p;
*p = (*p)->next;
free(t);
z++;
}
else
p = &(*p)->next;
}
return z;
}
```

```
void obrisi_listu(node **p){
while(*p != NULL) {
node *t = (*p)->next;
free(*p);
*p = t;
}
}
```

# Main

```
int main()
{
int quit = 0;
// lista
node *first = NULL; // lista utakmica
// fajl sa rezultatima
FILE *f;
char fn[MAXN];
// ime tima
char name[MAXN];
while(!quit) {
printf("\nOpcije:\n");
printf("[1] učitavanje podataka iz fajla\n");
printf("[2] ispis svih utakmica iz liste\n");
printf("[3] ispis svih utakmica sa neresenim rezultatom\n");
printf("[4] brisanje utakmica gosta\n");
printf("[5] ispis poena za tim\n");
printf("[6] izlaz iz programa\n\n");
char opcija = getch();
```

# Switch са функцијама

```
switch(opcija) {  
case '1':  
    // ako su rezultati vec procitani, prvo  
    obrisi postojeću listu  
    if(first != NULL)  
        obrisi_listu(&first);  
    printf("Ulazni fajl: ");  
    gets(fn);  
    f = fopen(fn, "r");  
    procitaj_fajl(f, &first);  
    fclose(f);  
    break;  
case '2':  
    ispis(first);  
    break;  
case '3':  
    ispis_neresenih(first);  
    break;  
case '4':  
    printf("Tim: ");
```

```
    gets(name);  
    printf("Brisano utakmica: %d\n",  
        brisanje_poena_gosta(&first,name));  
    break;  
case '5':  
    printf("Tim: ");  
    gets(name);  
    printf("Broj poena: %d\n",  
        broj_poena_za_tim(first,name));  
    break;  
case '6':  
    obrisi_listu(&first);  
    quit = 1;  
    break;  
}  
printf("\n");  
}  
return 0;  
}
```

ХВАЛА НА ПАЖЊИ!