

## Домаћи задатак јануар 2021. (PJISP)

Из задате улазне датотеке учитати податке о тв серијама у спрегнуту листу (SL) при чему податке о тв серији чине:

- назив (једна реч, до 20 карактера)
- број епизода (једна реч, до 20 карактера)
- година када је објављена прва епизода (цео број)
- просечна дужина трајања једне епизоде (реалан број)
- скраћена ознака земље порекла (једна реч, тачно 3 карактера)
- жанр (једна реч, до 20 карактера)

Након тога:

- а) За задату државу израчунати просечан број епизода по тв серији и тај податак сачувати у фајлу (користити `%.3lf` формат спецификатор) који у називу има ознаку те државе са екстензијом `.txt` и префиксом `serije_`
- б) У SL пронаћи серију који има највећу укупну дужину трајања (број епизода \* просечна дужина трајања једне епизоде) и податке о тој серији исписати на стандардни излаз.

Називи улазне датотеке, ознака државе, назив тв серије, задата година као и задати жанр се прослеђују као аргументи позива програма.

### Пример позива прогтама:

```
./zadatak serije.txt USA Blackadder 1995 Drama
```

Са податком `drzava = USA` и задатим улазом у датотецу `serije.txt`

```
TheSopranos 86 1999 55 USA Drama
OnlyFoolsAndHorses 64 1981 30 GBR Sitcom
Friends 236 1994 21 USA Sitcom
SrecniLjudi 70 1993 45 SRB Comedy
PorodicnoBlago 62 1998 55 SRB Drama
Blackadder 24 1983 30 GBR Comedy
LaPiovra 48 1984 105 ITA Drama
HouseMD 177 2004 42 USA Drama
```

**Пример решења:**

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_STRING      20 + 1
#define MAX_POREKLO     3 + 1

typedef struct serija_st {
    char naziv[MAX_STRING], brEpizoda[MAX_STRING],
        madeIn[MAX_POREKLO], zanr[MAX_STRING];
    int godinaPremijere;
    double avgTrajanje;
    struct serija_st *sledeci;
} SERIJE;

unsigned main__test_case(const int, char **);

void init_list(SERIJE **);
SERIJE *create_item(const char *, const char *, const int, const double, const
char *, const char *);
void add_to_list(SERIJE **, SERIJE *);
void load_data(FILE *, SERIJE **);
FILE *fopen_dat(char *, char *);
void item_print(FILE *, SERIJE *);
void avg_trajanje_state_criteria(FILE *, SERIJE *, const char *);
void max_duzina_trajanja(SERIJE *);
void brisanje_liste(SERIJE **);

void main(int argn, char **args)
{
    main__test_case(argn, args);
}

unsigned main__test_case(const int argn, char **args)
{
    if (argn != 6) {
        printf("USAGE: %s IN_FILENAME DRZAVA SERIJA GODINA ZANR\n", args[0]);
        exit(11);
    }
    SERIJE *g;

    char *izl = (char *) malloc( (10 + 1) * sizeof(char));
    char *extension = (char *) malloc(strlen("SCC.txt"));
    // SCC - SPECIFIC COUNTRY CODE

    strcpy(extension, (args[2])); strcat(extension, ".txt");
    strcpy(izl, "serije_"); strcat(izl, extension);
    free(extension);

    FILE *in = fopen_dat(args[1], "r");
    FILE *out = fopen_dat(izl, "w");

    init_list(&g);
    load_data(in, &g);
    avg_trajanje_state_criteria(out, g, args[2]);
    max_duzina_trajanja(g);
    brisanje_liste(&g);
}

```



```

FILE *fopen_dat(char *name, char *mode)
{
    FILE *f = fopen(name, mode);

    if(f == NULL)
    {
        printf("\nFile doesn't exist!\tError code: %s\n\n", "0xN0F1LE");
        exit(45);
    }
    return f;
}

void item_print(FILE *out, SERIJE *g)
{
    if(g == NULL)
    {
        printf("\nLista je prazna!\n");
        exit(7);
    }

    fprintf(out, "%-10s %-8s %d %5.2lf %-3s %-10s\n", g -> naziv,
                                                    g -> brEpizoda,
                                                    g -> godinaPremijere,
                                                    g -> avgTrajanje,
                                                    g -> madeIn,
                                                    g -> zanr);
}

void avg_trajanje_state_criteria(FILE *out, SERIJE *g, const char *madeIn)
{
    double suma = 0.0;
    unsigned counter = 0;

    while(g != NULL) {
        if((strcmp(g -> madeIn, madeIn) == 0)) {
            suma += atof((g -> brEpizoda));
            counter++;
        }
        g = g -> sledeci;
    }

    if(counter == 0) {
        fprintf(out, "Za %s ne postoje snimljene SERIJE!\n", madeIn);
        return;
    }
    else
    {
        fprintf(out, "Za %s prosečan broj epizoda je: %.3lf\n", madeIn,
                                                            suma / counter);

        return;
    }
}

void max_duzina_trajanja(SERIJE *g)
{
    SERIJE *tmp;
    tmp = NULL;
    double *max = (double *) malloc(sizeof(double)),
            *chk = (double *) malloc(sizeof(double));

```

```
while(g != NULL)
{
    if(tmp == NULL) {
        tmp = g;
        *max = atof(tmp -> brEpizoda) * (double) (tmp -> avgTrajanje);
    }
    *chk = atof(g -> brEpizoda) * (double) (g -> avgTrajanje);

    if(*chk > *max) {
        tmp = g;
        *max = atof(tmp -> brEpizoda) * (double) (tmp -> avgTrajanje);
    }
    g = g -> sledeci;
}
item_print(stdout, tmp);

free(chk);
free(max);
}

void brisanje_liste(SERIJE **glava)
{
    if(*glava == NULL)
        return;

    brisanje_liste(&((*glava) -> sledeci));

    free(*glava);
    (*glava) = NULL;
}
```