# Strukture podataka i algoritmi 1
# (zadatak - max 30 poena)

Jun, 2023

U koordinatnoj ravni tačke su određene parom koordinata, a geometrijske figure nizom tačaka. Za svaku tačku učitava se njena oznaka (niz karaktera) i njene koordinate (dva realna broja), a za svaku geometrijsku figuru oznaka figure (niz karaktera), broj temena figure (ceo broj) i za svako teme oznaka tačke. Ukoliko se kao oznaka temena unese nepostojeća tačka, teme se ignoriše. Ukoliko figura nema ni jednu tačku uklanja se iz spiska figura. Za svaku figuru se određuje obim figure, a za svaku tačku koliko figura sadrži tu tačku kao teme. Nakon učitavanja tačaka i figura, na ekranu se ispisuju sve figure sa njihovim obimima i tačka koja je teme najvećeg broja figura. Nakon toga se unosi oznaka tačke koja se briše iz spiska tačaka, pri čemu je potrebno obristi i sve figure kod kojih je uneta tačka teme.
**(4 poena, obavezno)**

Za rešavanje problema napisati sledeće funkcije:

a) Definisati sve potrebne složene tipove podataka neophodne za rešavanje opisanog problema.
**(3 poena, obavezno)**
b) Napisati funkciju **UcitajTacke** koja iz datoteke *Tacke.txt* učitava podatke o tačkama i formira listu/niz tačaka.
**(3 poena, obavezno)**
c) Napisati funkciju **NadjiTacku** koja za datu oznaku vraća pokazivač na traženog tačku.
**(2 poena)**
d) Napisati funkciju **UcitajFigure** koja iz datoteke *Figure.txt* učitava podatke o figurama i formira listu/niz figura.
**(3 poena + 2 poena)**
e) Napisati funkciju **ObimFigure** koja za prosleđenu oznaku figure određuje obim te figure.
**(4 poena)**
f) Napisati funkciju **MaxTacka** koja određuje tačku koja se javlja kao teme u najvećem broju figura. Ukoliko ima više tačka sa istom vrednošću vratiti bilo koju odnjih.
**(3 poena)**
g) Napisati funkciju **ObrisiTackuFiguru** koja za prosleđenu oznaku tačke realizuje brisanje te tačke i svih figura gde se ta tačka pojavjuje kao teme.
**(3 poena + 3 poena)**

Pri učitavanju podataka podrazumevati da su svi podaci korektno zadati.
Dozvoljeno je proširivanje struktura i definisanje novih, kao i definisanje drugih funkcija.
Maksimalan broj poena se dobija samo u slučaju da postoji veza između struktura koje čuvaju tačke i struktura koje čuvaju fugure, tj. ako postoje pokazivači "na jednu" ili "na obe strane".

Zadatak se može rešavati korišćenjem povezanih lista, nizova ili kombinacijom.
**Zadatak rešiti bez korišćenja globalnih promenljivih i bez unapred definisanih dužina korišćenih nizova** (izuzetak može da bude pomoćni niz za učutavanje stringova)**.**
Za funkcije koje nisu definisane mora da postoji deklaracija funkcije i kratak opis koji deo problema funkcija treba da reši i mogu se pozivati kao da postoji korektna definisana funkcija.

Rastojanje između tačaka $A(x_1, y_1), B(x_2, y_2)$ $d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

## Rešenje studenta

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

#define MAX 255

typedef struct Tacka{
    char *oznaka;
    float x;
    float y;
    int brojFigura;
    struct Tacka *sledeci;
} Tacka;

typedef struct Teme{
    Tacka *tacka;
    struct Teme *sledeci;
} Teme;

typedef struct Figura{
    char *oznaka;
    int brojTemena;
    Teme *temena;
    float obim;
    struct Figura *sledeci;
} Figura;

Tacka *UcitajTacke()
{
    FILE *file = fopen("Tacke.txt", "r");
    Tacka *glava = NULL;
    while (!feof(file))
    {
        Tacka *nova = (Tacka *)malloc(sizeof(Tacka));

        char buffer[MAX];
        fgets(buffer, MAX, file);
        buffer[strlen(buffer) - 1] = '\0';
        nova->oznaka = (char *)malloc((strlen(buffer) + 1) * sizeof(char));
        strcpy(nova->oznaka, buffer);
        fscanf(file, "%f%f\n", &nova->x, &nova->y);
        nova->brojFigura = 0;

        nova->sledeci = glava;
        glava = nova;
    }
    fclose(file);
    return glava;
}

Tacka *NadjiTacku(char *oznaka, Tacka *glava)
{
    while (glava != NULL && strcmp(oznaka, glava->oznaka))
        glava = glava->sledeci;
    return glava;
}

float Rastojanje(Tacka *A, Tacka *B)
{
    return sqrt(pow(A->x - B->x, 2) + pow(A->y - B->y, 2));
}
```

```
void DodajTeme(Teme **glava, Teme *novi)
{
    if (*glava == NULL) *glava = novi;
    else
    {
        Teme *temp = *glava;
        while (temp->sledeci != NULL)
            temp = temp->sledeci;
        temp->sledeci = novi;
    }
}

Figura *UcitajFigure(Tacka *glavaT)
{
    FILE *file = fopen("Figure.txt", "r");
    Figura *glavaF = NULL;
    while (!feof(file))
    {
        Figura *novi = (Figura *)malloc(sizeof(Figura));
        novi->sledeci = NULL;
        novi->temena = NULL;
        novi->obim = 0;

        char buffer[MAX];
        fgets(buffer, MAX, file);
        buffer[strlen(buffer) - 1] = '\0';
        novi->oznaka = (char *)malloc((strlen(buffer) + 1) * sizeof(char));
        strcpy(novi->oznaka, buffer);

        fscanf(file, "%d\n", &novi->brojTemena);
        int i;
        for (i = 0; i < novi->brojTemena; i++)
        {
            fgets(buffer, MAX, file);
            buffer[strlen(buffer) - 1] = '\0';
            Teme *novo = (Teme *)malloc(sizeof(Teme));
            novo->sledeci = NULL;

            Tacka *t = NadjiTacku(buffer, glavaT);
            novo->tacka = t;
            if (novo->tacka == NULL)
            {
                free(novo);
                novi->brojTemena--;
                i--;
            }
            else
            {
                DodajTeme(&novi->temena, novo);
                t->brojFigura++;
            }
        }

        if (glavaF == NULL) glavaF = novi;
        else
        {
            Figura *temp = glavaF;
            while (temp->sledeci != NULL)
                temp = temp->sledeci;
            temp->sledeci = novi;
        }
    }
    fclose(file);
    return glavaF;
```

```
}

Figura *NadjiFiguru(Figura *glava, char *oznaka)
{
    while (glava != NULL && strcmp(glava->oznaka, oznaka))
        glava = glava->sledeci;
    return glava;
}

void ObrisiFiguru(Figura **glava, Figura *obrisi)
{
    if (*glava == obrisi) *glava = (*glava)->sledeci;
    else
    {
        Figura *temp = *glava;
        while (temp->sledeci != obrisi)
            temp = temp->sledeci;
        temp->sledeci = obrisi->sledeci;
    }
    free(obrisi);
}

void ObrisiTacku(Tacka **glava, Tacka *obrisi)
{
    if (*glava == obrisi) *glava = (*glava)->sledeci;
    else
    {
        Tacka *temp = *glava;
        while (temp->sledeci != obrisi)
            temp = temp->sledeci;
        temp->sledeci = obrisi->sledeci;
    }
    free(obrisi);
}

void ObimFigure(char *oznakaF, Figura **glavaF)
{
    Figura *f = NadjiFiguru(*glavaF, oznakaF);

    Teme *prvo = f->temena;
    if (prvo == NULL)
    {
        ObrisiFiguru(glavaF, f);
        return;
    }
    if (prvo->sledeci == NULL)
    {
        f->obim = 0;
        return;
    }

    Teme *poslednje = NULL;
    Teme *temp = prvo;
    while (temp->sledeci != NULL)
    {
        f->obim += Rastojanje(temp->tacka, temp->sledeci->tacka);
        temp = temp->sledeci;
    }

    poslednje = temp;
    if (f->brojTemena > 2)
        f->obim += Rastojanje(poslednje->tacka, prvo->tacka);
}

void MaxTacka(Tacka *glava)
```

```
{
    Tacka *maxTacka = glava;
    int brojMax = glava->brojFigura;

    Tacka *temp = glava;
    while (temp != NULL)
    {
        if (temp->brojFigura > brojMax)
        {
            brojMax = temp->brojFigura;
            maxTacka = temp;
        }
        temp = temp->sledeci;
    }
    printf("%s - %d\n", maxTacka->oznaka, maxTacka->brojFigura);
}

void ObrisiTackuFiguru(char *oznakaT, Figura **glavaF, Tacka **glavaT)
{
    Tacka *t = NadjiTacku(oznakaT, *glavaT);

    Figura *temp = *glavaF, *pom;
    while (temp != NULL)
    {
        int obrisana = 0;
        Teme *teme = temp->temena;
        while (teme != NULL)
        {
            if (teme->tacka == t)
            {
                pom = temp;
                temp = temp->sledeci;
                ObrisiFiguru(glavaF, pom);
                obrisana = 1;
                break;
            }
            teme = teme->sledeci;
        }

        if (!obrisana)
            temp = temp->sledeci;
    }

    ObrisiTacku(glavaT, t);
}

void IspisiFigure(Figura *glava)
{
    while (glava != NULL)
    {
        printf("%s - %f\n", glava->oznaka, glava->obim);
        glava = glava->sledeci;
    }
}

int main()
{
    Tacka *glavaT = UcitajTacke();
    Figura *glavaF = UcitajFigure(glavaT);

    Figura *tempF = glavaF, *pomF;
    while (tempF != NULL)
    {
        pomF = tempF;
        tempF = tempF->sledeci;
```

```
        ObimFigure(pomF->oznaka, &glavaF);
    }

    IspisiFigure(glavaF);
    MaxTacka(glavaT);

    char oznakaB[MAX];
    fgets(oznakaB, MAX, stdin);
    oznakaB[strlen(oznakaB) - 1] = '\0';
    ObrisiTackuFiguru(oznakaB, &glavaF, &glavaT);
    IspisiFigure(glavaF);
}
```