

## Operacije sa vektorima

1. Napisati funkciju `void ZbirVektora (float x1, float y1, float z1, float x2, float y2, float z2, float* v)` koja računa zbir  $\vec{v} = \vec{u} + \vec{w}$  vektora  $\vec{u} = (x_1, y_1, z_1)$  i  $\vec{w} = (x_2, y_2, z_2)$ .
2. Napisati funkciju `void MnozenjeSkalarom (float x, float y, float z, float lambda, float* v)` koja računa vektor  $\vec{v} = \lambda \vec{u}$ , gde je  $\vec{u} = (x, y, z)$ .
3. Napisati C program koji za unetih  $n$  vektora i  $n$  skalara formira njihovu linearnu kombinaciju.  
*Napomena: Koristiti funkcije ZbirVektora, MnozenjeSkalarom.*
4. Napisati funkciju `float SkalarniProizvod (float x1, float y1, float z1, float x2, float y2, float z2)` koja računa skalarni proizvod vektora  $\vec{u} = (x_1, y_1, z_1)$  i  $\vec{w} = (x_2, y_2, z_2)$ .
5. Napisati funkciju `float NormaVektora (float x, float y, float z)` koja računa normu zadatog vektora  $\vec{v} = (x, y, z)$ .
6. Napisati funkciju `float Norma2Vektora (float x, float y, float z)` koja računa kvadrat norme datog vektora  $\vec{v} = (x, y, z)$ .
7. Napisati funkciju `float UgaoIzmedjuVektora (float x1, float y1, float z1, float x2, float y2, float z2)` koja računa ugao  $\angle(\vec{u}, \vec{w})$  izmedju vektora  $\vec{u} = (x_1, y_1, z_1)$  i  $\vec{w} = (x_2, y_2, z_2)$ . Funkcija treba da vrati vrednost ugla izraženu u stepenima (uzeti da je  $\pi = 3.14159265$ ).
8. Napisati funkciju `int LinearnoNezavisni (float x1, float y1, float z1, float x2, float y2, float z2)` koja za vektore  $\vec{u} = (x_1, y_1, z_1)$  i  $\vec{w} = (x_2, y_2, z_2)$  proverava da li su linearno nezavisni i vraća 1 ako jesu, a 0 ako nisu.
9. Napisati funkciju `float PovrsinaTrougla (float* A, float* B, float* C)` koja računa površinu  $\triangle ABC$  zadatog svojim temenima  $A(x_1, y_1, z_1)$ ,  $B(x_2, y_2, z_2)$ ,  $C(x_3, y_3, z_3)$ .
10. Napisati funkciju `int CetvrtoTeme (float* A, float* B, float* C, float* D)` koja računa koordinate tačke  $D$  takve da je četvorougao  $ABCD$  paralelogram. Funkcija vraća 0 ako su tačke  $A(x_1, y_1, z_1)$ ,  $B(x_2, y_2, z_2)$  i  $C(x_3, y_3, z_3)$  kolinearne, a 1 inače.
11. Napisati funkciju `int Kvadrat (float* A, float* C, float* B, float* D)` koja računa koordinate tačaka  $B$  i  $D$  takvih da je četvorougao  $ABCD$  kvadrat. Funkcija vraća 0 ako su tačke  $A(x_1, y_1, z_1)$  i  $C(x_3, y_3, z_3)$  identične, a 1 inače.
12. Napisati C program koji za tri unete tačke  $A, B$  i  $C$  određuje tačke  $D, F$  i  $G$  takve da je  $ABCD$  paralelogram, a  $ADFG$  kvadrat. Zatim, program računa površinu  $\triangle BDG$  i dužinu stranice kvadrata.  
*Napomena: Koristiti funkcije CetvrtoTeme, Kvadrat, PovrsinaTrougla.*
13. Napisati funkciju `void Ortogonalizacija (float* u, float* v, float* w)` čiji su argumenti 3 linearno nezavisna vektora  $\vec{u}$ ,  $\vec{v}$ ,  $\vec{w}$ . Funkcija treba da Gram-Schmidt-ovim postupkom odredi vektore  $\vec{u}_1$ ,  $\vec{v}_1$ ,  $\vec{w}_1$  koji čine ortonormiranu bazu prostora generisanog vektorima  $\vec{u}$ ,  $\vec{v}$ ,  $\vec{w}$ .
14. Napisati C program koji iz skupa  $n$  unetih vektora izdvaja linearno nezavisne. Ako dobijeni skup ima:
  - 1 element - program treba da izračuna njegovu normu;
  - 2 elementa - program treba da odredi ugao izmedju tih vektora i vektor normalan na njih;
  - 3 elementa - program treba da odredi ortonormiranu bazu prostora generisanog tim vektorima.

*Napomena: Koristiti funkcije LinearnoNezavisni, NormaVektora, VektorskiProizvod, Ortogonalizacija.*

## Afine transformacije

- Napisati funkciju `void Translacija (Tacka T, Tacka v)` koja računa sliku tačke  $T$  pri translaciji za vektor  $v$ .
  - Napisati funkciju `void Rotacija (Tacka T, Tacka Q, double alfa)` koja računa sliku tačke  $T$  pri rotaciji za ugao  $\alpha$  oko tačke  $Q$ .
  - Napisati funkciju `void Homotetija (Tacka T, Tacka Q, double lambda1, double lambda2)` koja računa sliku tačke  $T$  pri istezanju u pravcu koordinatnih osa sa koeficijentima  $\lambda_1, \lambda_2$  i centrom  $Q$ .
  - Napisati funkciju `void Smicanje (Tacka T, double lambda)` koja računa sliku tačke  $T$  pri smicanju sa koeficijentom  $\lambda$  u pravcu  $x$  ose.
  - Napisati C program koji kao ulaz (preko tastature) ima dve tačke  $A$  i  $C$ , koje predstavljaju koordinate donjeg levog i gornjeg desnog temena kvadrata (tačke  $B$  i  $D$  se određuju iz uslova da je  $ABCD$  kvadrat), i transformacije koje je potrebno izvršiti. Program treba da nacrtava sliku kvadrata pri svakoj transformaciji.
- Napisati funkciju `int AfinoPreslikavanje (Tacka A1, Tacka B1, Tacka C1, Matrica M)` koja određuje jednačine afinog preslikavanja koje slika kvadrat sa temenima  $A(-1, -1)$ ,  $B(1, -1)$ ,  $C(1, 1)$ ,  $D(-1, 1)$  u paralelogram  $A_1B_1C_1D_1$  (tačku  $D_1$  treba odrediti kao četvrto teme paralelograma). Funkcija vraća 1 ako preslikavanje postoji, a 0 ako ne postoji. Matrica  $M$  je dimenzije  $3 \times 3$  i funkcija u nju treba da upiše odgovarajuće koeficijente afinog preslikavanja.
  - Napisati funkciju koja određuje sliku jediničnog kruga upisanog u kvadrat  $ABCD$ .
  - Napisati C program koji poziva funkcije pod a) i b). Korisnik sa ulaza zadaje tri tačke, a program treba da ispiše jednačinu slike kruga pri ovoj afinoj transformaciji. Takođe, treba nacrtati kvadrat i u njega upisani krug, kao i njihove slike pri zadatoj transformaciji.  
*Napomena: Pogledati primer 4.1 u skripti.*

## Tačka, prava, ravan

- Napisati funkciju `void PresekPravih (Tacka P, Tacka pp, Tacka Q, Tacka qq, Tacka* X)` koja kao argumente uzima prave  $p$  i  $q$ , određene tačkom i vektorom pravca. Funkcija treba da vrati 0 ako su prave paralelne, 1 ako je presek jedinstvena tačka  $X$  i 2 ako se prave poklapaju.
- Napisati funkciju `void PresekDuzi (Tacka A, Tacka B, Tacka C, Tacka D, Tacka* X)` koja kao argumente uzima krajnje tačke dveju duži  $AB$  i  $CD$ , a vraća celobrojnu vrednost, i to 0 ako nema preseka, 1 ako je presek jedinstvena tačka  $X$  i 2 ako se duži seku u više od jedne tačke (tj. njihov presek je duž).
- Napisati funkciju `void Normala (Tacka M, Prava P, Prava* n)` koja računa normalu  $n$  iz tačke  $M$  na pravu  $p$ .
- Napisati funkciju `void SimetralaUgla (Tacka A, Tacka B, Tacka C, Prava* s)` koja računa simetralu  $s$  ugla  $\angle ABC$  sa temenom  $B$ .
- Napisati funkciju `float Rastojanje (Tacka M, Prava p)` koja računa rastojanje tačke  $M$  od prave  $p$ .
- Napisati C program koji kao ulaz (preko tastature) ima tri tačke  $A$ ,  $B$  i  $C$  date svojim koordinatama. Program treba da nacrtava  $\triangle ABC$ , kao i upisani, opisani i spolja pripisane krugove trougla.  
*Napomena: Koristiti funkcije PresekPravih, Normala, SimetralaUgla, Rastojanje.*
- Deklarisana je promenljiva `presek` kao niz tačaka dužine 2, tj. sa `Tacka presek[2]`. Napisati funkciju `int PresekPravaKrug (Krug k, Prava p, Tacka presek)` koja presečne tačke kruga  $k$  i prave  $p$  upisuje u niz tačaka (tj. `T[0] = presek1`, `T[1] = presek2`, ako postoje). Celobrojna vrednost funkcije treba da bude 0, 1 ili 2 u zavisnosti od broja presečnih tačaka.
- Napisati funkciju `void PravaPodUglom (Prava p, Tacka Q, float alfa, Prava* q)` koja vraća pravu  $q$  koja sadrži tačku  $Q$ , a sa datom pravom  $p$  zaklapa dati orijentisani ugao  $\alpha$  dat u stepenima.

9. Napisati funkciju `void PravaNaRastojanju (Prava p, float d, Prava q)` koja vraća niz od dve prave  $q_1$  i  $q_2$  koje su paralelne pravoj  $p$  i na rastojanju  $d \neq 0$  od nje.
10. Napisati funkciju `int Tangenta(Tacka A, Krug k, Prava tangenta)` koja vraća tangente na krug  $k$  iz tačke  $A$ . Funkcija treba da vrati broj tangenti.
11. Napisati funkciju `int PresekKrugova (Krug k1, Krug k2, Tacka presek)` koja upisuje presečne tačke krugova  $k_1$  i  $k_2$  u niz tačaka `presek`. Funkcija treba da vrati broj presečnih tečaka ta dva kruga.
12. Napisati funkciju `int Krug2tackeR (Tacka A, Tacka B, float r, Krug k)` koja vraća niz krugova koji sadrže tačke  $A$  i  $B$  i imaju poluprečnik  $r$ . Funkcija treba da vrati broj takvih krugova (najviše 2).
13. Napisati funkciju `int KrugTangentaTackaR (Prava t, Tacka A, float r, Krug k)` koja vraća niz krugova koji sadrže tačku  $A$ , dodiruju pravu  $t$  i imaju poluprečnik  $r$ . Funkcija treba da vrati broj takvih krugova.
14. Napisati funkciju `int ZajednickeTangente (Krug k1, Krug k2, Prava tangenta)` koja vraća niz zajedničkih tangenti krugova  $k_1$  i  $k_2$  (može ih biti najviše 4). Funkcija treba da vrati broj tangenti.

## Poligoni

1. Napisati funkciju `int Prost (Tacka* pol, int n)` koja vraća 1 ako je poligon prost, a 0 ako nije. Poligon je dat kao niz od  $n$  tačaka.
2. Napisati funkciju `int Unutar (Tacka A, Tacka* pol, int n)` koja za datu tačku  $A$  i zadat kao niz tačaka dužine  $n$  proverava da li je tačka unutar poligona. Funkcija vraća 1 ako je tačka unutar poligona, 0 ako pripada poligonu, a -1 ako je tačka izvan njega.
3. Napisati funkciju `Triangulisi (Tacka* pol, int n, Tacka* triangulacija)` koja računa triangulaciju poligona od  $n$  tačaka. Funkcija treba da vrati 0 ako je došlo do greške (npr. ako poligon nije prost), a broj trouglova inače. Niz od  $n - 2$  trougla triangulacije (tj. niz od  $3(n - 2)$  tačke) treba upisati na mesto pokazano pokazivačem `*triangulacija`.
4. Napisati funkciju `void KonveksniOmotac (Tacka tniz, int n, Tacka omotac)` koja za niz tačaka `tniz` dužine  $n$  računa konveksni omotač i upisuje ga u niz tačaka `omotac`. Funkcija treba da koristi:  
a) spori algoritam      b) brzi algoritam.

## Krive u parametarskom obliku

1. Napisati funkciju `int NacrtajElipsu (float a, float b)`. Argumenti funkcije su realne vrednosti koje predstavljaju dužine veće i manje poluose parametarski zadate elipse. Funkcija treba da nacrtaja elipsu i vraća 1 ako je nacrtana uspesno, a 0 inače.

## Bézier-ove krive

1. Napisati funkciju `int Bezier2 (Tacka t1, Tacka t2, Tacka t3, float nizx, float nizy)`. Argumenti funkcije su tri kontrolne tačke Bézier-ove krive drugog reda. Funkcija treba da vrati 0 ako Bézier-ova kriva ne postoji (tj. te tri tačke su kolinearne), a 1 ako ta kriva postoji. Funkcija treba da upiše u nizove `nizx` i `nizy` koeficiente Bézierovih polinoma stepena 2 za  $x$  i  $y$  koordinatu krive. Nizovi su dužine 3.
2. Napisati funkciju `int Bezier3 (Tacka t1, Tacka t2, Tacka t3, Tacka t4, float nizx, float nizy)`. Argumenti funkcije su četiri kontrolne tačke. Funkcija treba da vrati 0 ako Bézier-ova kriva ne postoji (tj. te tačke  $t_1, t_2, t_4$  ili  $t_1, t_3, t_4$  su kolinearne), a 1 ako ta kriva postoji. Funkcija treba da upiše u nizove dužine 4 `nizx` i `nizy` koeficijente Bézier-ovih polinoma stepena 3.
3. Napisati funkciju `int TackaNaKrivoj (Tacka tniz, int n, float u, Tacka B)` koja određuje tačku  $B(u)$  na Bézier-ovoj krivoj stepena  $n$  za zadato  $u$ . Argumenti funkcije su niz kontrolnih tačaka `tniz`, stepen krive  $n$  i realna vrednost  $u \in [0, 1]$ . Funkcija računa tačku  $B$  na krivoj koristeći De Casteljaou-ov algoritam i vraća 0 ako je došlo do greške, a 1 inače.

4. Napisati funkciju `int PodelaKrive (Tacka tniz, int n, float u, Tacka niz1, Tacka niz2)` koja deli datu Bézier-ovu krivu  $n$ -tog reda na dve Bézier-ove krive  $n$ -tog reda praveći rez u tački  $B(u)$  za dato  $u$ . Argumenti funkcije su niz kontrolnih tačaka `tniz` dužine  $n + 1 > 2$  i realna vrednost  $u \in [0, 1]$ . Funkcija treba da upiše u nizove `niz1` i `niz2` nove kontrolne tačke Bézier-ovih krivih. Funkcija vraća 0 ako je došlo do greške, a 1 inače.
5. Napisati funkciju `int PovecajStepenKrive (Tacka tniz, int n, int m, Tacka noviniz)` koja povećava stepen Bézier-ove krive za  $m$  bez promene same krive. Argumenti funkcije su niz kontrolnih tačaka `tniz`, stepen krive  $n$  i novi stepen krive  $m$ . Funkcija računa nove kontrolne tačke i upisuje ih u niz `noviniz`.
6. Napisati C program koji kao ulaz (preko tastature) ima celobrojnu vrednost  $n$  koja predstavlja stepen Bézier-ove krive, niz kontrolnih tačaka `tniz` i realnu vrednost  $u \in [0, 1]$ . Program treba da nacрта Bézier-ovu krivu i na njoj tačku  $B(u)$  za dato  $u$ , a zatim i nove kontrolne tačke koje odgovaraju krivama dobijenim iz početne pravljenjem reza u tački  $B(u)$ .  
*Napomena: Koristiti funkcije `TackaNaKrivnoj`, `PodelaKrive`.*
7. Napisati C program koji kao ulaz (preko tastature) ima celobrojnu vrednost  $n$  koja predstavlja stepen Bézier-ove krive, niz kontrolnih tačaka `tniz` i celobrojnu vrednost  $m$ . Program treba da nacрта Bézier-ovu krivu i njene kontrolne tačke, a zatim i nove kontrolne tačke koje se dobijaju kada se stepen krive povisi za  $m$ .  
*Napomena: Koristiti funkciju `PovecajStepenKrive`.*