

Laboratorijske vježbe 8.

1. Napisati strukturu "VektorN" koja predstavlja N-dimenzionalni vektor. Struktura treba da ima sljedeće parametre: dimenziju vektora i pokazivač na niz tipa double u kojem se čuvaju koordinate niza:

Potrebno je implementirati sljedeće funkcije koje rade sa strukturom VektorN:

```
VektorN KreirajVektorN(int dimenzija);
VektorN Zbir(VektorN v1, VektorN v2);
VektorN Razlika(VektorN v1, VektorN v2);
VektorN SkalarniProizvod(VektorN v1, VektorN v2);
double Norma(VektorN v1);
void Ispisi(VektorN v1) const;
```

Funkcija KreirajVektorN treba da kreira objekat tipa VektorN sa datom dimenzijom, te da alokira dinamički niz sa datom dimenzijom i popuni elemente niza sa nulama, te ako dođe problema koda alokacije memorije da se baci izuzetak. Funkcije Zbir, Razlika i SkalarniProizvod trebaju za data dva vektora da izračunaju njihov zbir, razliku te skalarni proizvod, te da odgovarajuće vrijednosti vrata kao rezultat, pri čemu treba baciti izuzetak ako su vektori različitih dimenzija. Funkcija Norma treba da vrati normu vektora, dok funkcija Ispisi treba da ispiše dati vektor u obliku $\{x_1, x_2, \dots, x_n\}$.

Potrebno je napraviti testni program u kojem će se demonstrirati upotreba datih funkcija za manipulaciju sa objektima tipa VektorN

2. Potrebno je sljedeći zadatak koji glasi:

"Napisati program koji od korisnika traži da unese dimenzije kvadratne matrice n , a zatim da unese dvije matrice formata $n \times n$ (pretpostaviti da su elementi matrica realni brojevi). Program nakon toga treba da ispiše zbir dvije unesene matrice. U programu treba da se nalaze sljedeće funkcije: "KreirajMatricu", "UnesiMatricu", "SaberMatrice", "IspisiMatricu" i "UnistiMatricu". Funkcija "KreirajMatricu" prima kao parametre dvojni pokazivač koji služi za pristup dinamički kreiranoj matrici (u nastavku ćemo ovaj pokazivač prosto zvati *dinamička matrica*), kao i dimenziju matrice n . Funkcija treba da alokira prostor za matricu formata $n \times n$, i dodijeli adresu alociranog prostora pokazivaču koji se koristi za pristup njenim elementima. Funkcija "UnesiMatricu" popunjava matricu elementima unesenim sa tastature, a prima kao parametre dinamičku matricu i dimenziju n . Funkcija "SaberMatrice" prima kao parametre dvije dinamičke matrice i dimenziju n . Ova funkcija treba da *kreira* novu dinamičku matricu (pozivom funkcije "KreirajMatricu"), da je popuni zbirom dvije dinamičke matrice koje su joj proslijeđene kao parametri, i da vrati kao rezultat dvojni pokazivač koji služi za pristup elementima novokreirane matrice. Funkcija "IspisiMatricu" kao parametre prima dinamičku matricu i dimenziju n , a ispisuje elemente matrice na ekran. Konačno, funkcija "UnistiMatricu" unistava dinamički kreiranu matricu koja joj se prosljeđuje kao parametar (zajedno sa dimenzijom n). Funkcija "KreirajMatricu" treba da baci izuzetak u slučaju da kreiranje ne uspije. Pri tome, ova funkcija mora da vodi računa da u slučaju da dođe do bacanja izuzetka "počisti iza sebe" sve uspjele alokacije, tako da ne dođe do curenja memorije. Bačeni izuzetak treba hvatati u glavnom programu. Obavezno testirati slučaj kada alokacija ne uspijeva (unosom prevelikog broja n).",

uraditi koristeći strukturu Matrica koja će da sadrži tri parametra: *broj_kolona*, *broj_vrsta* i dvostruki pokazivač *elementi* na tip *double* u kojem će se čuvati elementi matrice uz odgovarajuću prepravku prototipova traženih funkcija.