

Zadaci za 1. laboratorijske vježbe

1. Napišite funkciju čiji je prototip:

void ispis (int *polje, int n) ;

koja kao argumente prima pokazivač na početak cjelobrojnog polja (**polje**) i broj članova polja (**n**) te rekurzivno ispisuje članove polja od zadnjeg prema prvome.

Primjer: za polje [0, 1, 2, 3, 4] treba biti ispisano: 4 3 2 1 0

Napišite glavni program koji će učitati broj elemenata jednodimenzionalnog polja **n**. Dinamički alocirajte prostor za cjelobrojno polje **A** od **n** članova. Ako se ne može alocirati prostor za polje **A**, potrebno je ispisati poruku „Nema dovoljno memorije.“ i završiti s izvođenjem programa.

Ako je memorija uspješno alocirana, onda napunite polje **A** tako da vrijedi **A[i] = i** za svaki **i = 0, ..., n-1**. Članove polja ispišite od zadnjeg prema prvome korištenjem funkcije **ispis**.

2. Napišite funkciju čiji je prototip:

int zbrojiParne (int *polje, int n) ;

koja kao argumente prima pokazivač na početak cjelobrojnog polja (**polje**) i broj članova polja (**n**) te rekurzivno zbraja parne članove polja.

Primjer: za polje [0, 1, 2, 3, 4] funkcija treba vratiti 6.

Napišite glavni program koji će učitati broj elemenata jednodimenzionalnog polja **n**. Dinamički alocirajte prostor za cjelobrojno polje **A** od **n** članova. Ne treba provjeravati je li memorija uspješno alocirana za polje **A**.

Potrebno je napuniti polje **A** tako da vrijedi **A[i] = i** za svaki **i = 0, ..., n-1** te ispisati zbroj parnih članova polja (pozvati funkciju **zbrojiParne**).

3. Napišite funkciju čiji je prototip:

double pi (int n) ;

koja kao argument prima broj članova reda (**n**) te rekurzivno računa broj π prema izrazu:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

Primjer: za $n = 1$, funkcija treba vratiti 4; za $n = 2$, funkcija treba vratiti 2.666667; za $n = 10$, funkcija treba vratiti 3.041840, itd.

Napišite glavni program koji će učitati cijeli broj **n** te ispisati vrijednost broja π izračunatog korištenjem funkcije **pi** za zadani broj članova reda **n**.

4. Napišite funkciju čiji je prototip:

double f (double x, int n);

koja kao argument prima realni broj **x** i cijeli broj **n** te u vremenu $O(n)$ rekurzivno računa

izraz: $\frac{x^n}{n!}$

Primjer: za $x = 1$ i $n = 0$, funkcija treba vratiti 1; za $x = 1$ i $n = 1$, funkcija treba vratiti 1; za $x = 1$ i $n = 3$ funkcija treba vratiti 0.166667; itd.

Napišite glavni program u kojem će se učitati cijeli broj **n** i realni broj **x** te ispisati vrijednost koju vraća funkcija **f** za zadane vrijednosti **x** i **n**.

5. Napišite funkciju čiji je prototip:

int binarnoTrazi (int *polje, int n, int x) ;

koja kao argumente prima pokazivač na početak cjelobrojnog uzlazno sortiranog polja (**polje**), broj članova polja (**n**) te cijeli broj **x**. U funkciji postupkom binarnog pretraživanja treba provjeriti nalazi li se **x** u polju. Funkcija vraća indeks elementa **x**, ako se **x** nalazi u polju, a -1 inače.

Napišite glavni program u kojem će biti definirano jednodimenzionalno polje **A** s cjelobrojnim elementima 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 te cijeli broj **x = 4**.

Za broj **x** provjeriti nalazi li se u polju **A** (koristite funkciju **binarnoTrazi**). Potrebno je ispisati indeks člana polja, ako je **x** pronađen u polju **A**, a poruku „Broj se ne nalazi u polju.“, ako **x** nije pronađen u polju **A**.

6. Napišite funkciju čiji je prototip:

char *malaSlova (char *niz) ;

koja kao argument prima pokazivač na početak znakovnog niza (**niz**) te vraća pokazivač na početak novog znakovnog niza za koji je dinamički alocirana memorija u funkciji. Novi niz treba sadržavati iste znakove kao i ulazni niz, osim u slučaju velikih slova engleske abecede koja trebaju biti zamijenjena malim slovima.

Primjer: za ulazni znakovni niz "asp_i_ASP", funkcija treba vratiti pokazivač na novi niz "asp_i_asp".

Napišite glavni program u kojem će biti definiran znakovni niz (varijabla **niz**) sadržaja "asp_i_ASP" te ispisati znakovni niz koji je rezultat poziva funkcije **malaSlova** s argumentom **niz**.

7. Napišite funkciju koja prima pokazivač na polje cijelih brojeva i koja vraća pokazivač na novo polje koje se sastoji od kvadriranih elemenata ulaznog polja.
Primjerice, ako se ulazno polje sastoji od elemenata 1, 2, 3, 4 i 5, izlazno polje će biti 1, 4, 9, 16 i 25.

Potrebno je napisati i glavni program koji od korisnika učitava ulazni niz. Glavni program zatim poziva funkciju i ispisuje rezultate izvođenja funkcije.

Napomena: rezervaciju memorije u funkciji treba ostvariti pozivom funkcija malloc/realloc.

8. Napišite funkciju koji prima niz znakova koji se sastoji od slova engleske abecede, interpunkcijskih znakova i praznina. Funkcija stvara novi niz koji se dobiva izbacivanjem svih praznina iz izvornog niza.

Primjerice, za ulaz „Sunce nam dolazi!“, funkcija vraća niz „Suncenamdolazi!“.

Potrebno je napisati i glavni program koji od korisnika učitava znakovni niz. Glavni program zatim poziva funkciju i ispisuje rezultat izvođenja funkcije.

Napomena: rezervaciju memorije u funkciji treba ostvariti pozivom funkcija malloc/realloc.

9. Napišite **rekurzivnu** funkciju koja kao parametar prima pokazivač na rezervirani memorijski prostor cijelih brojeva i njegovu veličinu. Prototip funkcije je:

void f (int* polje, int n);

Funkcija treba rezervirani prostor popuniti rastućim vrijednostima koje su potencije broja 2, na način da element na indeksu 0 ima vrijednost 2^0 , na indeksu 1 vrijednost 2^1 itd.

Potrebno je napisati i glavni program koji od korisnika učitava cijeli broj **n** te zatim rezervira memorijski prostor za cjelobrojno polje od **n** članova. Glavni program zatim poziva funkciju **f** za to cjelobrojno polje i broj članova polja **n** i ispisuje rezultate izvođenja funkcije.

Napomena: rezervaciju memorije treba ostvariti pozivom funkcija malloc/realloc.

10. Napišite **rekurzivnu** funkciju čiji je prototip:

double f (double z, int k);

koja kao argument prima realni broj **z** i cijeli broj **k** te u vremenu $O(k)$ računa izraz:

$$\frac{(-1)^k z^{2k+1}}{(2k+1)!}$$

U glavnom programu je potrebno definirati realni broj **z** = 0.5 te funkciju **f** pozivati za različite vrijednosti broja **k** i ispisivati rezultate poziva.