

Osnovi programiranja (Matematika)

I popravni kolokvijum

18.1.2021.

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.c**.

1. Napisati C program koji za učitano celobrojnu vrednost promenljive **n** ($n > 0$) računa vrednost funkcije $f(n)$ na sledeći način:

$$f(n) = \sum_{i=1}^n \frac{i^{2i-1} + (2i)^n}{(2n)!} \cdot (1 + 3 + 5 + \dots + (2i - 1))$$

Ulaz sadrži samo jedan ceo broj **n**. Izlaz predstaviti kao **realan broj u formatu na 10 mesta sa 3 decimale**.

NAPOMENA: Funkcije **abs/fabs**, **pow**, kao i **svoje funkcije** ne smete koristiti ni kod uslova definisanosti, ni kod računanja vrednosti.

Takođe, tekući član sume **OBAVEZNO** računati na osnovu prethodnog člana, ukoliko je to moguće.

Ulaz	Izlaz
2	4.208
3	6.150
4	8.497

2. Kamion za isporuke ekspres službe ima zadatak da svakodnevno isporučuje pošiljke do adresa naručilaca. Svakog jutra, kamion najpre dolazi do skladišta u kome radnici prenose pakete određene težine i zapremine do njegovog kontejnera, pakujući ih redom. Sam kamion je definisan maksimalnom zapreminom i nosivošću, tako da ne može da prihvati sve pakete. Potrebno je napisati program koji će ispisati ukupnu zapreminu i težinu svih paketa koji su tog dana spakovani u kamion i poslati na isporuku, kao i njihovu prosečnu težinu.

U prvoj liniji standardnog nalazi se nosivost kamiona **N** (realan broj) i zapremina njegovog kontejnera u okviru kog se pakuju paketi **Z** (realan broj). Zatim se u svakoj novoj liniji standardnog ulaza unose redom: **širina**, **dužina**, **visina** i **težina** paketa (realni brojevi) koji je trenutno na redu za unos. Unošenje sa standardnog ulaza se prekida onog trenutka kada ukupna težina paketa premaši nosivost kamiona ili ukupna zapremina paketa premaši dozvoljenu zapreminu kontejnera. Na standardnom izlazu je potrebno ispisati, u okviru jednog reda, **ukupnu zapreminu i težinu** paketa koji su uneti u kamion, kao i njihovu **prosečnu težinu**. Izlaze formatirati u obliku sa **2 decimale**.

Za realizaciju programa, potrebno je napisati i sledeću pomoćnu funkciju:

- **ZapreminaPaketa (float sirina, float duzina, float visina)** – koja za prosleđene dimenzije paketa računa i vraća kao rezultat njegovu zapreminu.

Ulaz	Izlaz	Ulaz	Izlaz
100 150 5 5 6 12 4 2 6 20	150.00 12.00 12.00	100 150 1 2 5 2 4 2 1 3 4 1 2 5 5 5 5 10	26.00 10.00 3.33

Broj poena: 1. zadatak – 10 poena, 2. zadatak – 16 poena
Vreme izrade: 120 minuta

Osnovi programiranja (Matematika)

II popravni kolokvijum

18.1.2021.

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.c**.

1. Napisati program koji:

- sadrži funkciju **UcitajMat** koja za matricu celih brojeva date dimenzije $m \times n$ učitava elemente matrice po vrstama.
- sadrži funkciju **StampajMat** koja za matricu celih brojeva date dimenzije $m \times n$ štampa matricu u matičnom obliku po vrstama.
- sadrži funkciju **TransformisiNiz** koja prosleđeni niz dužine n transformiše tako što na mestu svakog člana originalnog niza postavlja broj njegovog pojavljivanja u celokupnom nizu.
Primer: [5 1 5 6 8 6 2] => [2 1 2 2 1 2 1]
- sadrži funkciju **TransformisiMat** koja transformiše prosleđenu matricu dimenzije $m \times n$ tako da svaka njena vrsta sadrži brojeve pojavljivanja originalnog elemenata u okviru te vrste. Transformaciju svake vrste realizovati pomoću prethodno definisane funkcije **TransformisiNiz**.

U glavnom delu programa najpre se učitavaju pozitivni prirodni brojevi **M** i **N** ($1 \leq M, N \leq 30$) koji predstavljaju redom dimenzije matrice, a zatim se koristeći funkciju **UcitajMat** unose elementi matrice. Koristeći funkciju **TransformisiMat** izmeniti datu matricu. Odštampati dobijenu matricu koristeći funkciju **StampajMat**. Na kraju, odštampati broj jedinstvenih elemenata svake vrste transformisane matrice.

Ulaz: M = 5, N = 5					Izlaz				
5	4	1	1	5	2	1	2	2	2
11	2	11	5	3	2	1	2	1	1
6	5	5	4	4	1	2	2	2	2
5	9	2	4	3	1	1	1	1	1
8	8	1	1	1	2	2	3	3	3
					1	3	1	5	0

2. Napisati program u kome se najpre unose nazivi ulaznog i izlaznog fajla, a zatim se iz ulaznog fajla iščitavaju podaci o košarkašima. Za svakog košarkaša se pamte sledeće informacije:

- ime** (jedna reč),
- prezime** (jedna reč),
- pozicija** (jedna reč – PG, SG, SF, PF, C),
- visina** (realan broj).

Napisati sledeće funkcije:

- Funkciju **UcitajKosarkase** koja iz datoteke datog naziva učitava redom podatke o košarkašima, pri čemu se svi podaci o jednom košarkašu nalazi u zasebnom redu, odvojeni razmakom. Podaci se učitavaju dok se ne dođe do kraja datoteke, a kao povratnu vrednost ova funkcija treba da vrati broj učitanih košarkaša.
- Funkciju **IspisKosarkasa** koja za prosleđeni niz košarkaša, dužinu niza i naziv izlazne datoteke, u okviru izlazne datoteke prosleđenog naziva ispisuje u zasebnim redovima podatke o košarkašima u obliku: [ime] [prezime] ([pozicija]) – [visina]m“.
- Funkciju **IzdvojIgracelstePozicije** koja za prosleđenu poziciju (PG – Point Guard, SG – Shooting Guard, SF – Small Forward, PF – Power Forward ili C – Center) kreira novi niz košarkaša koji sadrži samo one igrače koji igraju na toj poziciji. Funkcija kao povratnu vrednost vraća broj igrača na prosleđenoj poziciji.

- Funkciju **SortirajPoVisini** koja prosleđeni niz košarkaša sortira po visini igrača, počevši od najvišeg.

U glavnom delu programa učitati najpre nazive ulazne i izlazne datoteke, a zatim učitati i željenu poziciju po kojoj se vrši pretraga igrača. Zatim:

- Pomoću funkcije **UcitajKosarkase** učitati niz košarkaša iz ulazne datoteke.
- Pomoću funkcije **IzdvojlgracelstePozicije** kreirati novi niz košarkaša koji igraju na odgovarajućoj, u startu učitanoj, poziciji.
- Pomoću funkcije **SortirajPoVisini** sortirati igrače koji igraju na odgovarajućoj poziciji po visini.
- Pomoću funkcije **IspisKosarkasa** ispisati igrače sa učitane pozicije, sortirane po visini od najvišeg do najnižeg.

ulaz.txt	Izlaz.txt (pozicija: PG)
Victor Oladipo SG 1.93 Lebron James SF 2.06 Kyrie Irving PG 1.88 Andre Drummond C 2.11 Chris Paul PG 1.85 Kemba Walker PG 1.83 James Harden SG 1.96 Klay Thompson SG 1.98 Kevin Durant SF 2.08 Blake Griffin PF 2.06 Kristaps Porzingis PF 2.21 Dirk Nowitzki PF 2.13 Marc Gasol C 2.11 Stephen Curry PG 1.91	Stephen Curry (PG) - 1.91m Kyrie Irving (PG) - 1.88m Chris Paul (PG) - 1.85m Kemba Walker (PG) - 1.83m

Broj poena: 1. zadatak – 20 poena, 2. zadatak – 20 poena
Vreme izrade: 120 minuta