

Osnovi programiranja I kolokvijum

14.11.2020.

I grupa

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**. Od dva ponuđena zadatka birate **jedan** koji ćete raditi.

Napisati C program koji:

1. Za učitane celobrojnu vrednost promenljive **n** računa vrednost funkcije $f(n)$ na sledeći način:

- ukoliko je uneti broj **n** pozitivan funkcija $f(n)$ se dobija kao rezultat date sume:

$$f(n) = \sum_{i=1}^n (5 + 9 + 13 + \dots + 4i + 1) \cdot \frac{(-1)^i \cdot (3n)!}{(3n - i)! \cdot i^{2i-1}}$$

- ukoliko je uneti broj **n** negativan ili jednak nuli, vrednost funkcije $f(n)$ se računa na sledeći način:

Svaku cifru broja **n** treba povećati za 1. Ukoliko je trenutna cifra vrednosti 9, treba je zameniti nulom. Na primer:

-8920 -> -9031

-928 -> -39

Ulaz sadrži samo jedan ceo broj **n**. Ukoliko je uneti broj **n** pozitivan (prvi slučaj), izlaz predstaviti kao **realan broj u formatu na 10 mesta sa 3 decimale**. U suprotnom, ukoliko je **n** jednako nuli ili negativno (drugi slučaj), izlaz predstaviti kao **ceo broj**.

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. Zadatak se boduje i delimično, ukoliko se reši samo jedan način računanja funkcije $f(n)$.

Ulaz	Izlaz
2	22.500
4	56.238
-8920	-9031
-928	-39

2. U luna parku "IMI land" najveću atrakciju predstavlja rolerkoster "INFO zmaj". Kapacitet rolerkosteru za jednu vožnju je ograničen brojem putnika i ukupnom težinom svih putnika. Kako je vožnja poprilično opasna, pristup rolerkosteru nije dozvoljen svima, već isključivo **punoletnim** posetiocima luna parka. Takođe, kako bi putnici bili sigurni, neophodno je da budu minimalne visine **160cm**. U redu za rolerkoster čeka određeni broj posetilaca. Na vožnju se mogu ukrcati putnici ukoliko njihova ukupna težina ne prelazi maksimalnu dozvoljenu i ako nije premašen broj dozvoljenih putnika. Napisati program koji određuje koji posetioci su uspeli da se ukrcaju na rolerkoster i koja je njihova ukupna težina i prosečna visina.

U prvoj liniji standardnog ulaza nalazi se broj posetilaca **N** (prirodan broj $1 \leq N \leq 50$). U narednoj liniji unosi se broj **M** (prirodan broj $1 \leq M \leq 50$) koji predstavlja maksimalan broj putnika u toku jedne vožnje, a zatim ceo broj **T** koji predstavlja maksimalnu ukupnu težinu putnika u toku vožnje. U svakoj narednoj liniji nalazi se broj godina, visina i težina posetioca koji je na redu (celobrojne vrednosti). Čim neki posetilac uspe da se ukrca na vožnju, ispisuju se podaci o njemu u zasebnom redu (redom kojim su uneti: godine, visina i težina). Na standardnom izlazu ispisati ukupnu težinu ukrvanih posetilaca (u kilogramima), kao i njihovu prosečnu visinu (u centimetrima, realna vrednost, **zaokružiti na dve decimale**).

Ulaz	Izlaz
6	19 182 76
4	23 179 90
300	23 175 93
16 175 70	259
19 182 76	178.67
21 159 55	

23 179 90	
23 175 93	
21 161 62	

Broj poena: 1. zadatak – 13 poena, 2. zadatak – 18 poena

Vreme izrade: 75 minuta

Osnovi programiranja I kolokvijum

14.11.2020.

II grupa

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**. Od dva ponuđena zadatka birate **jedan** koji ćete raditi.

Napisati C program koji:

3. Za učitano celobrojnu vrednost promenljive **n** računa vrednost funkcije $f(n)$ na sledeći način:

- ukoliko je uneti broj **n** pozitivan funkcija $f(n)$ se dobija kao rezultat date sume:

$$f(n) = \sum_{i=1}^n \frac{(-1)^{i-1} \cdot (4n)!}{(4n-i)! \cdot i^{4i-1}} \cdot (7 + 12 + 17 + \dots + 5i + 2)$$

- ukoliko je uneti broj **n** negativan ili jednak nuli, vrednost funkcije $f(n)$ se računa na sledeći način:

Svaku cifru broja **n** treba smanjiti za 1. Ukoliko je trenutna cifra vrednosti 0, treba je zameniti cifrom 9. Na primer:

-1715 -> -604

-7809 -> -6798

Ulaz sadrži samo jedan ceo broj **n**. Ukoliko je uneti broj **n** pozitivan (prvi slučaj), izlaz predstaviti kao **realan broj u formatu na 10 mesta sa 3 decimale**. U suprotnom, ukoliko je **n** jednako nuli ili negativno (drugi slučaj), izlaz predstaviti kao **ceo broj**.

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. Zadatak se boduje i delimično, ukoliko se reši samo jedan način računanja funkcije $f(n)$.

Ulaz	Izlaz
2	47.688
4	142.412
-1715	-604
-7809	-6798

4. U zgradi PMF-a, nakon godina čekanja, konačno je ugrađen lift. Na trećem spratu fakulteta, održavaju se konsultacije iz predmeta Osnovi programiranja. Grupa studenata kasni na konsultacije i želi da iskoristi lift kako bi što pre stigli. Zbog ograničenja lifta, ne mogu svi da stignu na konsultacije na vreme. U liftu je dozvoljeno da se nađe određeni broj osoba, ali je potrebno da ukupna težina svih osoba ne prelazi maksimalnu dozvoljenu nosivost lifta. Zbog trenutne epidemiološke situacije, na ulazu u fakultet, meri se temperatura studenata i pri tome, pristup zgradi nije dozvoljen studentima koji imaju temperaturu **višu od 37 stepeni**. Napisati program koji određuje koji su studenti stigli na vreme na konsultacije i odrediti njihovu prosečnu temperaturu i ukupnu težinu.

U prvoj liniji standardnog ulaza nalazi se broj studenata **N** (prirodan broj $1 \leq N \leq 50$). U narednoj liniji unosi se broj **M** (prirodan broj $1 \leq M \leq 50$) koji predstavlja maksimalan broj osoba u liftu, a zatim ceo broj **T** koji predstavlja maksimalnu nosivost lifta. U svakoj narednoj liniji nalazi se temperatura (realna vrednost) i težina studenta (celobrojna vrednosti). Čim neki student uspe da uđe u lift, ispisati podatke o njemu u zasebnom redu (temperatura – u formatu sa dve decimale i težina). Na standardnom izlazu ispisati ukupnu težinu studenata koji su se vozili liftom (u kilogramima), kao i njihovu prosečnu telesnu temperaturu (realna vrednost, **zaokružiti na dve decimale**).

Ulaz	Izlaz
6	36.50 76
4	36.20 90
300	37.00 93
38.2 70	259
36.5 76	36.57
38.1 55	

36.2 90	
37 93	
36.3 62	

Broj poena: 1. zadatak – 13 poena, 2. zadatak – 18 poena

Vreme izrade: 75 minuta