

# Osnovi programiranja I kolokvijum

12.11.2017.

## 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**, rešenje 3. zadatka **mora** da se nalazi u fajlu **Zadatak3.c**. Od tri ponuđena zadatka birate dva koja ćete raditi.

Napisati C program koji:

1. Od cifara datog prirodnog broja **n** formira nov broj **m** ali tako da nov broj sadrži svaku drugu cifru broja **n**, s desna na levo:

Primer:  $n=1234 \rightarrow m=42$ ;  $n=56781 \rightarrow m=175$

Ulaz sadrži samo jedan broj **n**, izlaz sadrži samo jedan broj **m** koji predstavlja nov broj i to u formatu **na deset mesta**.

2. Za unete **celobrojne** vrednosti za promenljive **x** i **n** računa vrednost funkcije  $f(x, n)$  na sledeći način:

- a. Ukoliko je uneti broj **n** pozitivan, funkcija  $f(x, n)$  ima vrednost date sume:

$$f(x, n) = \sum_{i=1}^n \frac{i^i + x^i + (x+i)!}{(1+2+\dots+i) + (1+2+\dots+n)}$$

- b. Ukoliko je uneti broj **n** negativan ili jednak nuli, funkcija  $f(x, n)$  se izračunava na sledeći način:

$$f(x, n) = \begin{cases} (x-n) + 10, & |x+n| \geq 10 \\ \frac{\max(|x+n|, |n-x|)}{\sqrt{\min(|x+n|, |x-n|)}}, & \text{inače} \end{cases}$$

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. Na ulazu se zadaje najpre vrednost **x** a zatim u novom redu vrednost **n**. Izlaz sadrži samo jedan broj koji predstavlja vrednost funkcije  $f(x, n)$  i to u formatu **na deset mesta sa pet decimale**. Ukoliko smatrate da funkcija na nekom delu nije definisana, štampajte samo **NIJE DEFINISANO**.

3. U maketarskom klubu „Plovila i letelice“ prilikom okupljanja članova, često se počaste nekim sokovima i slatkišima. Za tu namenu u klubu postoje dve kutije za priloge. U prvu kutiju se ubacuju prilozima do 100 dinara i iz te kutije se novac troši isključivo na kupovinu sokova. U drugu kutiju se ubacuju prilozima od 100 dinara i više i koriste se za kupovinu hrane. Dogovor je da ukoliko u prvoj kutiji ima više od **S** dinara, novac se ubacuje u drugu, bez obzira na to koliki je prilog, osim u slučaju ako u drugoj kutiji ima više od **H** dinara, kada se prilog ne dodaje uopšte. Napisati program koji vodi stanje novca u kutijama. Svako dodavanje priloga se označava kôdom 0, kupovina sokova ima kôd 1, a hrane kôd 2. Sok i hranu nije moguće kupiti ako u odgovarajućoj kutiji nema dovoljno novca. U program se na početku unose iznosi **S** i **H** i broj izmena **N** u stanju kutija (ukupan broj dodavanja i trošenja novca), a zatim se za svaku izmenu stanja unosi kôd i iznos u dinarima. Na kraju ispisati stanje u kutijama. Sve novčane vrednosti su celi brojevi.

Primer. Ulaz:  $S = 200$   $H = 1000$   $N = 6$  0 80 0 70 1 30 0 90 0 150 0 90

Izlaz: 210 240

Broj poena: 1. zadatak – 4 poena, 2. zadatak – 9 poena, 3. zadatak – 7 poena

Vreme izrade: 120 minuta

# Osnovi programiranja I kolokvijum

12.11.2017.

## 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**, rešenje 3. zadatka **mora** da se nalazi u fajlu **Zadatak3.c**. Od tri ponuđena zadatka birate dva koja ćete raditi.

Napisati C program koji:

4. Od cifara datog prirodnog broja **n** formira nov broj **m** ali tako da nov broj sadrži samo parne cifre broja **n**, s desna na levo:

Primer: n=1234 -> m=42; n=56781 -> m=86

Ulaz sadrži samo jedan broj **n**, izlaz sadrži samo jedan broj **m** koji predstavlja nov broj i to u formatu **na deset mesta**.

5. Za unete celobrojne vrednosti za promenljive **x** i **n** računa vrednost funkcije  $f(x, n)$  na sledeći način:

a. Ukoliko je uneti broj **n** pozitivan, funkcija  $f(x, n)$  ima vrednost date sume:

$$f(x, n) = \sum_{i=1}^n \frac{i^n + x^n + (x + 1 + i)!}{(2 + 4 + \dots + 2i) + 1 + n}$$

b. Ukoliko je uneti broj **n** negativan ili jednak nuli, funkcija  $f(x, n)$  se izračunava na sledeći način:

$$f(x, n) = \begin{cases} (x + n) + 5, & 5 \leq |x + n| \leq 10 \\ \frac{\sqrt{\max(|x + n|, |x - n|)}}{\min(|x + 3n^2|, n^4 - x)}, & \text{inače} \end{cases}$$

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. Na ulazu se zadaje najpre vrednost **x** a zatim u novom redu vrednost **n**. Izlaz sadrži samo jedan broj koji predstavlja vrednost funkcije  $f(x, n)$  i to u formatu **na deset mesta sa pet decimale**. Ukoliko smatrate da funkcija na nekom delu nije definisana, štampajte samo **NIJE DEFINISANO**.

6. U potrazi za blagom, Luka i drugari organizuju višesatni izlet. Proučavali su mapu i zaključili da, na osnovu terena, imaju nekoliko etapa koje treba da savladaju. Potrebno je da predvide koliko vremena će provesti na izletu ako znaju koliko je dugačka svaka etapa, kojom brzinom mogu da se kreću kroz taj deo terena i uračunaju da će nakon svake etape praviti pauzu, čija će dužina zavisiti od vremena potrebnog da pređu tu etapu. Procena je da im je za pauzu potrebno tri puta manje vremena od vremena provedenog u hodu. Napisati program u kome se unosi broj etapa koje Luka sa drugarima treba da pređe, a zatim za svaku etapu brzina kojom mogu da se kreću po tom terenu i dužina etape. Odrediti ukupno vreme u minutima potrebno da stignu do cilja, računajući i pauze posle svake etape, osim posle poslednje etape. Brzine kojom se kreću se unose u km/h, dužine etapa u metrima.

Primer. Ulaz: N=5 5 1500 3.8 800 4.8 1600 2.1 1200 5.9 300 Izlaz: 116.27

Broj poena: 1. zadatak – 4 poena, 2. zadatak – 9 poena, 3. zadatak – 7 poena

Vreme izrade: 120 minuta

# Osnovi programiranja I kolokvijum

12.11.2017.

## III grupa

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime\_BrIndeksa** 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**, rešenje 3. zadatka **mora** da se nalazi u fajlu **Zadatak3.c**. Od tri ponuđena zadatka birate dva koja ćete raditi.

Napisati C program koji:

7. Od cifara datog prirodnog broja **n** formira nov broj **m** ali tako da nov broj sadrži samo cifre koje su veće od prosečne vrednosti prethodnih cifara broja **n**, s desna na levo:

Primer: n=1524 -> m=45; n=39281 -> m=189

Ulaz sadrži samo jedan broj **n**, izlaz sadrži samo jedan broj **m** koji predstavlja nov broj i to u formatu **na deset mesta**.

8. Za unete celobrojne vrednosti za promenljive  $x$  i  $n$  računa vrednost funkcije  $f(x, n)$  na sledeći način:

a. Ukoliko je uneti broj  $n$  pozitivan, funkcija  $f(x, n)$  ima vrednost date sume:

$$f(x, n) = \sum_{i=1}^n (-1)^i \frac{x^{3n} + (3 + 6 + \dots + 3i)}{(3i)! + (3i)^n}$$

b. Ukoliko je uneti broj  $n$  negativan ili jednak nuli, funkcija  $f(x, n)$  se izračunava na sledeći način:

$$f(x, n) = \begin{cases} \frac{1}{(3x + 3n) - 10}, & 15 \geq |3x + 3n| > 10 \\ \frac{\max(|3x + 3n|, |3n - 3x|)}{\sqrt{\min(|3x + 3n|, |3x - 3n|)}}, & \text{inače} \end{cases}$$

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. Na ulazu se zadaje najpre vrednost  $x$  a zatim u novom redu vrednost  $n$ . Izlaz sadrži samo jedan broj koji predstavlja vrednost funkcije  $f(x, n)$  i to u formatu **na deset mesta sa pet decimale**. Ukoliko smatrate da funkcija na nekom delu nije definisana, štampajte samo **NIJE DEFINISANO**.

9. U maketarskom klubu „Plovila i letelice“, za izradu jednog od brodova, potrebna im je platnena traga određene širine i što duža. Komadi platna, koje imaju u klubu, su pravougaonog oblika i različitih dimenzija, tako je neophodno da ih sastavljaju što efikasnije da bi napravili traku potrebne dužine. Napisati program koji učitava broj komada platna koje maketari imaju na raspolaganju, zatim potrebnu širinu trake i željenu dužinu, a zatim se unose dimenzije komada platna sve dok se ne postigne željena dužina ili ne iskoriste svi komadi platna. Program treba da ispiše dužinu napravljene trake. Sve dimenzije se zadaju u centimetrima kao realni brojevi.

Primer. Ulaz: N = 6 S = 10 D = 200

22.5 30 40.8 15 8.78 15 33.7 5.8 7 9.6 46.1 45.9

Izlaz: 131.48

Broj poena: 1. zadatak – 4 poena, 2. zadatak – 9 poena, 3. zadatak – 7 poena

Vreme izrade: 120 minuta