

Intel 64 - zadaci za vežbu

Zadatak 1. Napisati asemblersku funkciju:

```
int secret(int x);
```

koja izračunava #tajni# broj datog celog broja x . #Tajni# broj datog broja x se dobija iterativnim postupkom u kome se polazi od datog broja kao tekuće vrednosti i u svakom koraku se sabiraju parovi susednih cifara tekuće vrednosti da bi se dobila jedna cifra naredne vrednosti (tako bi npr. ovako dobijene cifre od broja 1234 bile redom $1 + 2$, $2 + 3$ i $3 + 4$, odn. naredna vrednost prilikom izračunavanja njegovog #tajnog# broja bi bila 357). Ukoliko je zbir nekog para cifara veći od 9, onda se cifre zbira opet sabiraju da bi se dobila nova cifra u intervalu $[0, 9]$. Opisani iterativni postupak se ponavlja sve dok se ne dobije jednocifrena vrednost i ta vrednost je #tajni# broj polaznog broja. Napisati potom i C program koji učitava broj x , poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

8169

izlaz treba da bude:

2

Zadatak 2. Napisati asemblersku funkciju:

```
void diff(double * x, double * y, int n, double * r);
```

koja, koristeći paralelne SSE2 instrukcije, za nizove dužine n na koje pokazuju pokazivači x i y izračunava sumu $|x[i] - y[i]|$, za i od 0 do $n-1$. Rezultat smestiti na lokaciju na koju pokazuje r . Napisati potom i C program koji sa standardnog ulaza učitava n , alokira prostor za nizove, učitava elemente nizova, poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

3

0.45 1.2 -2.1

1.23 -2.31 1.24

izlaz treba da bude:

7.63

Zadatak 3. Napisati asemblersku funkciju:

```
int count_primes(int a, int b);
```

koja određuje koliko ima prostih brojeva u zatvorenom intervalu $[a, b]$ ($1 < a \leq b$). Napisati potom i C program koji sa standardnog ulaza učitava a i b , zatim poziva funkciju i ispisuje rezultat na standardnom izlazu. Na primer, za ulaz:

2 100

izlaz treba da bude:

25

Zadatak 4. Napisati asemblersku funkciju:

int takeuhi(int x, int y, int z);

koja izračunava Takeuhi-jevu funkciju, zadatu na sledeći način:

$$f(x, y, z) = \begin{cases} 1 + f(f(x-1, y, z), f(y-1, z, x), f(z-1, x, y)) & \text{za } y < x \\ z & \text{inače} \end{cases}$$

Napisati potom i C program koji testira ovu funkciju. Na primer, za unos oblika:

18 12 6

ispis programa treba da bude:

13

Zadatak 5. Napisati asemblersku funkciju:

*void center_of_mass(int n, float *P, float *m, float *C);*

koja, koristeći paralelne SSE instrukcije izračunava entar mase datog niza čestica. Broj čestica je $n > 0$, dok se koordinate čestica nalaze u nizu P (u njemu se najpre nalaze x, y i z koordinata prve čestice, a zatim tako redom i za sve ostale čestice). Vrednosti masa čestica su date u nizu m. Pretpostavlja se da su sve mase pozitivne. Centar mase je tačka čije se koordinate upisuju u niz C, a koje se izračunavaju po sledećoj formuli:

$$r = \frac{\sum_{i=0}^{n-1} m_i r_i}{\sum_{i=0}^{n-1} m_i} \quad (r \in \{x, y, z\})$$

Napisati potom i C program koji učitava najpre broj čestica, a zatim i koordinate i mase čestica (prvo se navedu koordinate prve čestice i njena masa, a zatim tako i za sve ostale čestice), nakon čega poziva funkciju i ispisuje koordinate centra mase na ekranu. Primer - za unos oblika:

3

1 2 0 1

1 0 1 2

0 0 2 3

rezultat treba da bude:

0.5 0.333333 1.33333

Zadatak 6. Napisati asemblersku funkciju:

int square_free(int x);

koja proverava da li je dati pozitivan ceo broj x kvadratno slobodan. Za broj x kažemo da je kvadratno slobodan ako za svaki prost broj p koji deli x broj p^2 ne deli x. Funkcija treba da vrati 1 ako je broj kvadratno slobodan, a 0 u suprotnom. Napisati potom i C program koji testira ovu funkciju. Na primer, za ulaz:

4040985

izlaz treba da bude:

1

Zadatak 7. Napisati asemblersku funkciju:

int perfect(int n);

koja za dati broj n ispituje da li je savršen. Broj je savršen ako je jednak zbiru svih svojih pravih delilaca. Funkcija vraća 1 ako je broj savršen, a 0 u suprotnom. Napisati potom i C program koji učitava broj n , poziva funkciju i ispisuje njen rezultat. Na primer, za ulaz:

8128

izlaz treba da bude:

1

Zadatak 8. Napisati asemblersku funkciju:

*int remove_elements(int *a, int n);*

koja modifikuje niz dužine n na koji pokazuje pokazivač a tako što iz njega uklanja sve elemente koji ne dele sumu prethodnih elemenata u polaznom nizu. Funkcija treba da vrati dužinu rezultujućeg niza. Napisati potom i C program koji testira ovu funkciju. Prostor za niz alocirati dinamički (u `main()` funkciji). Na primer, za ulaz:

6

3 4 1 9 8 5

izlaz treba da bude:

3 1 5

Zadatak 9. Napisati asemblersku funkciju:

*void fractions(int *f, int n, int *r);*

koja pronalazi najveći razlomak u nizu od n razlomaka na koji pokazuje f . Niz f je niz celih brojeva dužine $2n$ i u njemu se najpre nalaze brojilac i imenilac prvog razlomka, zatim brojilac i imenilac drugog razlomka, i tako dalje. Pretpostaviti da su svi imenioci pozitivni celi brojevi. Rezultat treba smestiti u niz dužine 2 na koji pokazuje r . Napisati potom i C -program koji učitava broj razlomaka n , zatim alocira prostor za niz razlomaka, učitava same razlomke, poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

4

4 15
-19 20
3 4
8 9

izlaz treba da bude:

8 9

Zadatak 10. Napisati asemblersku funkciju:

*int circles(int n, double * cr);*

koja, koristeći paralelne SSE2 instrukcije, za dati skup krugova u ravni ispituje koliko postoji parova krugova takvih da je jedan krug sadržan u drugom krugu. Krugovi su dati nizom na koji pokazuje cr , pri čemu se u nizu najpre nalaze x i y koordinata centra prvog kruga, a zatim i poluprečnik prvog kruga, nakon čega slede x i y koordinata centra drugog kruga, pa poluprečnik drugog kruga, itd. Broj krugova dat je parametrom n. Smatra se da je krug sadržan u drugom krugu čak i ako ga dodiruje sa unutrašnje strane. Napisati potom i C program koji sa standardnog ulaza učitava broj krugova, a zatim i podatke o krugovima (u gore opisanom redosledu), zatim poziva funkciju i na kraju ispisuje rezultat na standardni izlaz. Na primer, za ulaz:

5

1.0 1.0 2.0

0.1 0.1 1.0

2.0 1.0 1.0

2.5 3.5 3.0

1.0 4.5 5.0

izlaz treba da bude:

3

Zadatak 11. Napisati asemblersku funkciju:

unsigned reverse(unsigned x);

koja određuje ceo broj koji se dobija obrtanjem dekadnih cifara datog celog broja x. Napisati potom i C program koji testira datu funkciju. Na primer, za ulaz:

3546

izlaz treba da bude:

6453

Zadatak 12. Napisati asemblersku funkciju:

int powerful(int x);

koja proverava da li je dati pozitivan ceo broj x moćan. Za broj x kažemo da je moćan ako za svaki prost broj p koji deli x broj p^2 takođe deli x. Funkcija treba da vrati 1 ako je broj moćan, a 0 u suprotnom. Napisati potom i C program koji testira ovu funkciju. Na primer, za ulaz:

69148800

izlaz treba da bude:

1

Zadatak 13. Napisati asemblersku funkciju:

*void simpson(double a, double b, int n, double *r);*

koja koristeći paralelne SSE2 instrukcije izračunava integral funkcije $f(x) = x^3$ na intervalu $[a, b]$ korišćenjem Simpsonove formule:

$$\int_a^b f(x)dx \approx \frac{h}{3} \left(f(a) - f(b) + 4 \sum_{i=1}^n f(a + (2i-1)h) + 2 \sum_{i=1}^n f(a + 2ih) \right)$$

Funkcija prihvata granice intervala a i b , parametar n iz formule, kao i pokazivač na lokaciju na koju treba upisati rezultat. Napisati potom i C program koji testira ovu funkciju. Na primer, za ulaz:

1.2 4.6 1000

izlaz treba da bude:

111.418000

Zadatak 14. Napisati asemblersku funkciju:

int ackermann(int m, int n);

koja izračunava Ackermann-ovu funkciju:

$$A(m, n) = \begin{cases} n + 1 & \text{za } m = 0 \\ A(m - 1, 1) & \text{za } m > 0, n = 0 \\ A(m - 1, A(m, n - 1)) & \text{za } m > 0, n > 0 \end{cases}$$

Napisati potom i C program koji sa standardnog ulaza učitava m i n ($m, n \geq 0$), zatim poziva funkciju i ispisuje rezultat na standardnom izlazu. Na primer, za ulaz:

3 9

izlaz treba da bude:

4093

Zadatak 15. Napisati asemblersku funkciju:

*void quadratic_form(int n, float **a, float *x, float *f);*

koja koristeći paralelne SSE instrukcije izračunava vrednost kvadratne forme u datoj tački, tj. izračunava proizvod:

(

$$f = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Funkcija prihvata dimenziju matrice n, pokazivač na niz pokazivača na vrste matrice, pokazivač na niz koordinata vektora x, kao i pokazivač f na lokaciju na koju treba upisati rezultat. Napisati potom i C program koji testira ovu funkciju. Na primer, za ulaz:

5

1.2 2.3 3.4 1.1 4.0
1.6 0.2 6.4 2.0 1.1
3.3 3.2 2.7 2.8 5.4
2.1 4.4 3.5 2.7 4.7
0.9 3.0 1.0 2.1 4.5

1.0 1.1 1.2 1.3 1.4

izlaz treba da bude:

103.156998