

MIKRORAČUNARI - ISPIT - MART 2010.

1. (40 poena) Napisati *IA-32* asemblersku funkciju:

```
int periodic(int * a, int n);
```

koja ispituje da li je dati niz celih brojeva dužine n na čiji početni element pokazuje pokazivač a periodičan. Za niz kažemo da je periodičan ako postoji broj k ($1 \leq k \leq n/2$) takav da za svako i ($0 \leq i < n - k - 1$) važi $a[i + k] = a[i]$. Funkcija vraća dužinu najmanjeg perioda, odnosno 0 ako niz nije periodičan. Napisati potom i *C*-program koji učitava dužinu niza n , dinamički alocira prostor za celobrojni niz date dužine, zatim učitava elemente niza, poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

```
7
1 2 3 1 2 3 1
```

izlaz treba da bude:

```
3
```

2. (30 poena) Napisati *IA-32* asemblersku funkciju:

```
void linear3(double * a, double * b, double * x);
```

koja, koristeći matematički koprocessor (*FPU*), pomoću Kramerovog pravila izračunava rešenje sistema linearnih jednačina:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3\end{aligned}$$

Koeficienti a_{ij} dati su nizom realnih brojeva dužine 9 na čiji početni element pokazuje pokazivač a , tako što se u nizu najpre nalaze koeficienti prve jednačine, zatim druge i na kraju treće jednačine sistema. Koeficienti b_i dati su nizom realnih brojeva dužine 3 na čiji početni element pokazuje pokazivač b . Rešenje sistema se smešta u niz realnih brojeva dužine 3 na čiji početni element pokazuje pokazivač x . Pretpostaviti da sistem ima jedinstveno rešenje. Napisati potom i *C*-program koji sa standardnog ulaza učitava koeficiente sistema tako što najpre učitava koeficiente prve jednačine, zatim slobodni član prve jednačine, zatim koeficiente druge jednačine i njen slobodni član i na kraju koeficiente treće jednačine i njen slobodni član, zatim poziva funkciju i ispisuje rezultat na standardnom izlazu. Na primer, za ulaz:

```
1 1 1 6
1 2 3 14
1 1 -1 0
```

izlaz treba da bude:

1 2 3

3. (30 poena) Napisati *ARM* 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