

MIKRORAČUNARI - ISPIT - SEPTEMBAR 2010.

1. (30 poena) Napisati *IA-32* asemblersku funkciju:

```
int happy(int x);
```

koja ispituje da li je dati broj $x > 0$ srećan. Srećni brojevi se definišu na sledeći način: neka je data sekvenca brojeva x_0, x_1, x_2, \dots takva da je $x_0 = x$, a broj x_k je jednak zbiru kvadrata dekadnih cifara broja x_{k-1} , za $k > 0$. Ako se ova sekvenca završava brojem 1 (jer bi tada i svaki sledeći broj u sekvenci bio jednak 1), tada za početni broj x kažemo da je srećan. U suprotnom, ako se sekvenca ne završava (ne sadrži broj 1, već se periodično nastavlja u beskonačnost), tada broj x nije srećan. Može se dokazati da sekvence brojeva koji nisu srećni uvek sadrže broj 4. Napisati potom i *C*-program koji sa standardnog ulaza učitava x , zatim poziva funkciju i ispisuje rezultat na standardnom izlazu. Na primer, za ulaz:

167

izlaz treba da bude:

1

2. (40 poena) Napisati *IA-32* asemblersku funkciju:

```
void nth_root(double x, int n, double eps, double * r);
```

koja, koristeći *ne-paralelne SSE2* instrukcije, izračunava n -ti koren broja $x \geq 0$ ($n > 0$) pomoću sledećeg iterativnog postupka:

$$x_0 = 1$$
$$x_{k+1} = \frac{1}{n} \left[(n-1)x_k + \frac{x}{x_k^{n-1}} \right]$$

Za aproksimaciju n -tog korena uzima se prvo x_{k+1} takvo da je $|x_{k+1} - x_k| < \epsilon$ (pri čemu je vrednost ϵ data parametrom *eps* funkcije). Rezultat se upisuje na adresu na koju pokazuje pokazivač r . Napisati potom i *C*-program koji sa standardnog ulaza učitava redom x , n i *eps*, zatim poziva funkciju i ispisuje rezultat na standardnom izlazu. Na primer, za ulaz:

3.0

5

0.000001

izlaz treba da bude:

1.245731

3. (30 poena) Napisati *ARM* asemblersku funkciju:

```
void digits(int n, int *l, int *g);
```

koja određuje najveću i najmanju cifru dekadnog zapisa broja $n > 0$. Najveća cifra smešta se na lokaciju na koju pokazuje pokazivač g , dok se najmanja cifra smešta na lokaciju na koju pokazuje pokazivač l . Napisati potom i *C*-program koji učitava broj n , poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

56351

izlaz treba da bude:

1 6

NAPOMENA: Izrada zadataka traje 240 minuta.