

1. (35 poena) Napisati *IA-32* 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
```

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

```
double nth_root(double x, int n, double eps);
```

koja, koristeći matematički koprosesor (*FPU*), 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  $\epsilon$  data parametrom *eps* funkcije). 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. (35 poena) Napisati *ARM* asemblersku funkciju:

```
int nth_prime(int n);
```

koja određuje  $n$ -ti prost broj. Napisati potom i *C*-program koji učitava  $n$ , poziva funkciju i ispisuje njen rezultat na standardnom izlazu. Na primer, za ulaz:

```
35
```

izlaz treba da bude:

```
149
```