

1. (35 poena) Napisati *IA-32* asemblersku proceduru:

```
unsigned concat(unsigned x, unsigned y);
```

koja izračunava vrednost neoznačenog celog broja koji nastaje nadovezivanjem dekadnih zapisa brojeva  $x > 0$  i  $y > 0$ . Pretpostaviti da se tom prilikom neće javiti prekoračenje. Napisati potom i *C*-program koji učitava pozitivne cele brojeve  $x$  i  $y$ , poziva funkciju i ispisuje njen rezultat. Na primer, za ulaz:

```
123 45
```

izlaz treba da bude:

```
12345
```

2. (35 poena) Napisati asemblersku funkciju:

```
double exp_rational(int m, double *p, int n, double *q, double x);
```

koja koristeći matematički koprocesor (*FPU*) izračunava vrednost funkcije:

$$R(x) = \frac{p_0 + p_1 e^x + p_2 e^{2x} + \dots + p_m e^{mx}}{q_0 + q_1 e^x + q_2 e^{2x} + \dots + q_n e^{nx}}$$

Funkcija prihvata redom stepen polinoma broioca  $m$ , pokazivač  $p$  na niz koeficijenata polinoma broioca počev od koeficijenta najmanje težine, stepen polinoma imenioca  $n$ , pokazivač  $q$  na niz koeficijenata polinoma imenioca (opet počev od koeficijenta najmanje težine) kao i vrednost  $x > 0$  za koju se računa vrednost funkcije. Izračunata vrednost se vraća kao povratna vrednost funkcije. Napisati potom i *C*-program koji učitava ulazne podatke (u istom redosledu kao i u deklaraciji funkcije) i poziva funkciju, a zatim štampa rezultat na standardni izlaz. Prostor za nizove koeficijenata alocirati dinamički. Na primer, za ulaz:

```
2 2.1 3.2 1.2
3 0.3 1.1 5.1 4.2
3.4
```

izlaz treba da bude:

```
0.009994
```

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
```