# Uvod u programiranje

1. ispitni rok - 11. siječnja 2019.

Rješenja zadataka 1 - 4 napisati na vlastitim papirima i predati ih u košuljici.

Rješenja zadataka 5 - 7 napisati na za to predviđeno mjesto uz tekst zadatka te list sa zadacima predati u košuljici.

#### 1. (20 bodova)

Svaki zapis tekstne datoteke ulaz.txt sadrži jedan cijeli broj iz intervala [1, 99 999 999] i znak '#' na kraju retka. Broj zapisa u datoteci ulaz.txt nije ograničen. Načiniti novu tekstnu datoteku izlaz.txt koja redom, za svaki broj n iz intervala [1, 99 999 999] sadrži po jedan zapis u kojem se nalazi broj n i broj pojavljivanja toga broja u datoteci ulaz.txt.

Datoteka ulaz.txt smije se pročitati samo jednom, a u rješenju <u>nije dopušteno</u> korištenje agregatnih tipova podataka.

Primjer sadržaja ulaz.txt

```
984532#...

12#...

45987652#...

3625322#...

567821222#...

1#...

65321221#...
```

Primjer sadržaja izlaz.txt

#### 2. (18 bodova)

Struktura student\_s sadrži podatke o ispitima jednog studenta i sastoji se od matičnog broja (int), prezimena i imena studenta (40+1 znak) te podataka o 40 predmeta od kojih se svaki sastoji od šifre predmeta (int) i ocjene (short). Napisati funkciju pokNaNajveci koja će za zadano jednodimenzijsko polje struktura čiji su članovi tipa strukture student\_s vratiti pokazivač na onaj član polja koji sadrži podatke o ispitima studenta s najvećim prosjekom ocjena. Ako takvih (s jednakim najvećim prosjekom) ima više, vratiti pokazivač na član polja koji ima najveći indeks.

### 3. (17 bodova)

Napisati funkciju sumaReda tipa double. Funkcija treba izračunati sumu reda

$$1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} \dots$$

Izračun sume treba prekinuti kad novi članovi prestanu doprinositi sumi. U rješenju nije dopušteno koristiti funkcije ili programske odsječke koji nezavisno računaju potenciju ili faktorijel.

#### 4. (16 bodova)

Napisati program koji s tipkovnice učitava jedan nenegativni cijeli broj (unsigned int) i na zaslon ispisuje njegovu binarnu vrijednost. U rješenju nije dopušteno koristiti agregatne tipove podataka.

Primjeri izvršavanja programa

5. (3	B bo	da)
-------	------	-----

U prostor u okviru napisati koja je vrijednost upisana u varijablu tipa float čiji je heksadekadski prikaz FF800000.

## 6. (3 boda)

Uz pretpostavku da struktura tocka\_s sadrži član x tipa int, nadopuniti programski odsječak (isključivo iznad crte) tako da postane sintaktički ispravan.

```
struct tocka_s *p;
int x;
x = _____ p ____.x;
```

## 7. (3 boda)

Prikazati što će se tijekom izvršavanja odsječka programa ispisati na zaslonu.

```
int x[] = {7, 8};
printf("%d", *((char *)&x + 1));
```

```
RJEŠENJA
1.
#include <stdio.h>
#define N 99999999
int main(void) {
   int br;
   int n;
   FILE *ulaz, *izlaz, *tmp;
   ulaz = fopen("ulaz.txt", "r");
   tmp = fopen("tmp.bin", "w+b");
   while (fscanf(ulaz, "%d#", &br) == 1) {
      fseek(tmp, (long)(br - 1) * sizeof(int), SEEK_SET);
      if (fread(&n, sizeof(int), 1, tmp) == 1) {
         ++n;
      } else {
         n = 1;
      fseek(tmp, -1L * sizeof(int), SEEK_CUR);
      fwrite(&n, sizeof(int), 1, tmp);
   }
   fclose(ulaz);
   fseek(tmp, 0L, SEEK_SET);
   izlaz = fopen("izlaz.txt", "w");
   for (int i = 1; i <= N; ++i) {</pre>
      if (fread(&n, sizeof(int), 1, tmp) == 0) {
         n = 0;
      fprintf(izlaz, "%8d %d\n", i, n);
   }
   fclose(izlaz);
   fclose(tmp);
   return 0;
}
2.
typedef struct ocjena_s {
   int sifra;
   short ocjena;
} ocjena_predmeta;
typedef struct student_s {
   int maticni_broj;
   char ime_i_prezime[40+1];
   ocjena_predmeta ocjene[40];
} student;
```

```
student *pokNaNajveci(student *studenti, int n) {
   int ocjena, suma, nOcjena, indeksNajveceg = -1; // za slučaj da nitko nema niti jednu
ocjenu
   float prosjek, najveciProsjek = 0.f; // prosjek ne može biti negativan niti 0
   for (int i = 0; i < n; ++i) {
     suma = 0;
      nOcjena = 0;
      for (int j = 0; j < 40; ++j) {
         ocjena = (studenti+i)->ocjene[j].ocjena;
         if (ocjena > 0) {
            suma += ocjena;
            ++nOcjena;
         }
      if (nocjena > 0) {
         prosjek = suma / (float)nOcjena;
         if (prosjek >= najveciProsjek) {
            najveciProsjek = prosjek;
            indeksNajveceg = i;
         }
      }
   }
   return indeksNajveceg == -1 ? NULL : studenti + indeksNajveceg;
}
double sumaReda(int x) {
   double suma = 1., staraSuma, clan = 1.;
   int i = 1;
   do {
     staraSuma = suma;
     clan = -clan * x / i++;
      suma += clan;
   } while (suma != staraSuma);
   return suma;
}
4.
#include <stdio.h>
int main(void) {
   unsigned int n;
   printf("Upisati broj > ");
   scanf("%u", &n);
   for (int i = 31; i >= 0; --i){
     printf("%d", (n >> i) & 1);
   return 0;
}
```

6. y = (\*p).x;

5.

-∞

7. 0