

Programiranje 2 — tretji izpitni rok
27. avgust 2021

Ime in priimek: _____

Vpisna številka:

--	--	--	--	--	--	--	--

Točke (izpolnjujejo ocenjevalci):

1. naloga:

--

 / 30

2. naloga:

--

 / 30

3. naloga:

--

 / 40

Skupaj:

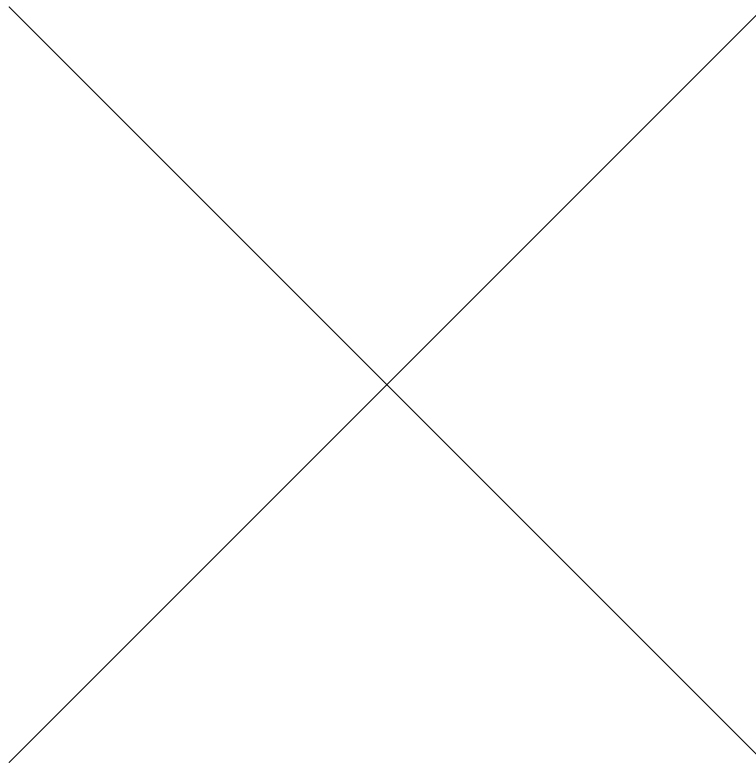
--

 / 100

Vsako nalogo rešujte samo na tisti strani, kjer je zapisano njeno navodilo!

Oddajte samo to polo (brez dodatnih listov)!

Veliko uspeha!



- ① V vsaki vrstici besedilne datoteke je zapisano celo število z intervala $[1, 10^9]$. V datoteki se skriva zaporedje: njegov prvi člen (imenujmo ga a_1) najdemo v prvi vrstici, drugi člen (a_2) je zapisan v vrstici z zaporedno številko $(1 + a_1)$, tretji (a_3) se nahaja v vrstici z zaporedno številko $(1 + a_1 + a_2)$ itd. Zaporedje se konča, ko vrstica, v kateri bi se moral nahajati naslednji člen, ne obstaja.

Napišite program, ki zaporedje iz podane vhodne datoteke prepíše v podano izhodno datoteko (vsak člen naj zapiše v svojo vrstico). Pot do vhodne datoteke je prvi, pot do izhodne datoteke pa drugi parameter ukazne vrstice.

Sledi primer vhodne in pripadajoče izhodne datoteke:

2
7
4
10
1
8
3
9

2
4
3

② Napišite sledečo funkcijo:

```
char** pretvori(char tabela[][N], int m)
```

V vsaki od m vrstic podane tabele je zapisan niz, ki se kot ponavadi konča z znakom z ASCII-kodo 0. Število N je definirano kot konstanta. Funkcija naj izdelata tabelo m kazalcev, vsak od njih pa naj kaže na začetek kopije istoležnega niza iz podane tabele. Vse izdelane tabele naj zasedajo samo toliko prostora, kot je nujno potrebno. Funkcija naj vrne kazalec na začetek izdelane tabele kazalcev.

- ③ Podanih je n oseb, označenih z indeksi od 0 do $n - 1$. Vsaka ima bodisi nič bodisi dva otroka. Osebe lahko predstavimo s strukturami takšnega tipa:

```
typedef struct _Oseba {
    int indeks;           // indeks osebe
    struct _Oseba* prviOtrok; // kazalec na otroka z manjšim indeksom oz. NULL,
                           // če oseba nima otrok
    struct _Oseba* drugiOtrok; // kazalec na otroka z večjim indeksom oz. NULL,
    } Oseba;              // če oseba nima otrok
```

Napišite sledečo funkcijo:

`Oseba* zgradi(int n, int* starsi)`

Kazalec `starsi` kaže na začetek tabele dolžine n , v kateri je na indeksu i zapisan indeks starša osebe z indeksom i . Natanko ena oseba je brez starša; zanjo je v tabeli zapisano število -1 . Funkcija naj izdela drevo, v katerem je vsaka oseba predstavljena s strukturo tipa `Oseba`, kazalci `prviOtrok` in `drugiOtrok` pa kažejo na otroke posameznih oseb. Funkcija naj vrne kazalec na koren tega drevesa (ta pripada osebi brez starša). Slika na desni prikazuje primer tabele in pripadajočega drevesa (znaki \times predstavljajo vrednosti `NULL`).

0	1	2	3	4	5	6	7	8
4	5	8	1	1	-1	4	8	5

