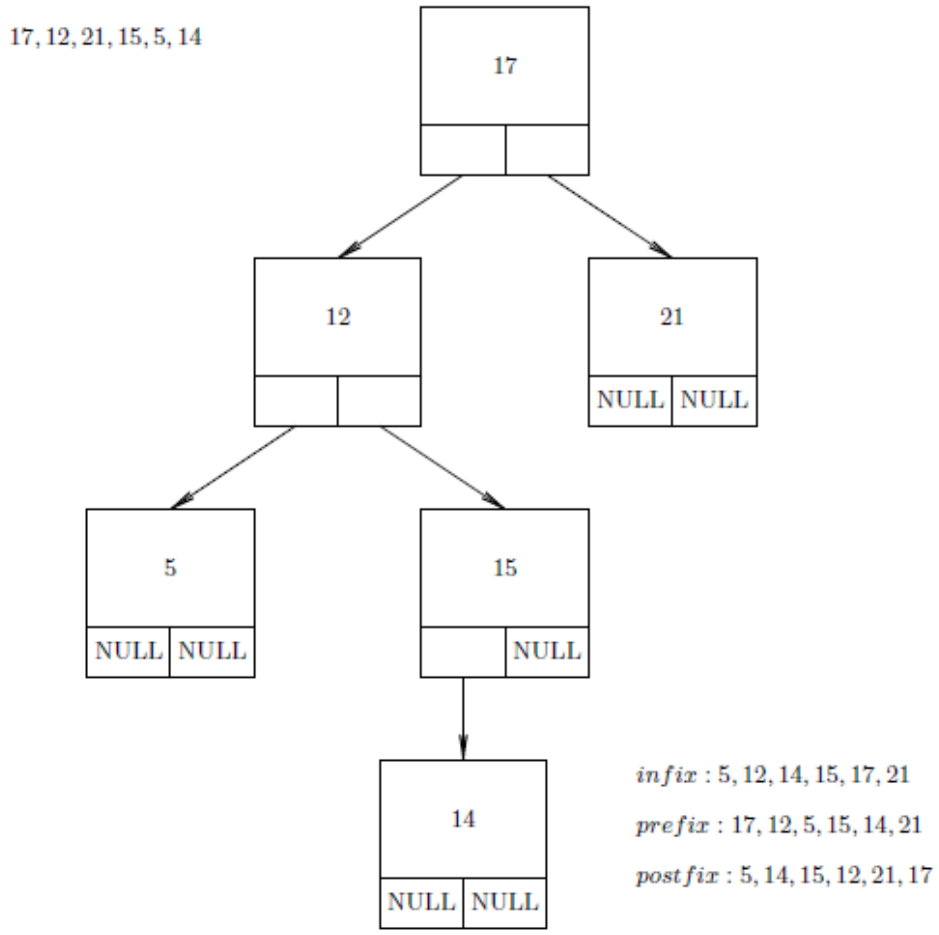
Zadatak 1: Obilazak Stabla

Vremensko ograničenje	Memorijsko ograničenje	ulaz	izlaz
0,5 s	32 MB	standardni ulaz	standardni izlaz

Napisati program koji učitava iz prvog reda standardnog ulaza broj čvorova stabla ($n<80$) i učitava u posebnim redovima inorder i preorder zapise (u vidu niski lkd i kld) jednog istog stabla nalazi i ispisuje na standardni izlaz njegov postorder zapis. Pretpostaviti da stablo ima do 80 čvorova označenih jednim ASCII znakom.

Ulaz	Izlaz
7 zstoipd iosztpd	ztsodpi

Pomoć:



INFIX, INORDER obilazak stabla = levo, koren, desno
PREFIX, PREORDER obilazak stabla = koren, levo, desno,
POSTFIX, POSTORDER obilazak stabla = levo, desno, koren

Nacrtati binarno stablo za koje INORDER (LKD) obilazak daje *zstoipd*, a PREORDER (KLD) obilazak *iosztpd*. Ključ čvora je slovo engleske abecede.

SKICA RESENJA:

1. KLD[0] mora biti koren stabla.

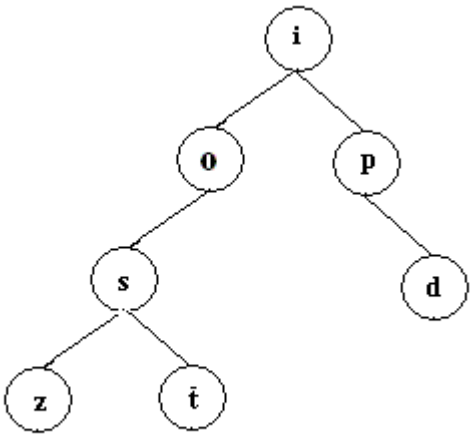
2. Ako je *n* ukupan broj čvorova i ako je *i* pozicija korena u nisci LKD, onda je

LKD[0..i-1] zapis levog podstabla u redosledu LKD i

LKD[i+1..n-1] zapis desnog podstabla u redosledu LKD.

3. Ako je *i* pozicija korena u nisci LKD, onda je KLD[1..i] zapis levog poddrveta u redosledu KLD (u levom poddrvetu je *i* cvorova). Slicno, KLD[i+1..n-1] je zapis desnog poddrveta u poretku KLD.

4. Rekurzivno primenimo pravila 1..3 (tj. KLD[1] je koren levog podstabla, KLD[i+1] je koren desnog stabla, i=4 jer LKD[4]=koren=*e*)



Zadatak 2

Asterix i Rimljani

Vremensko ograničenje
1.5 s
Memorjsko ograničenje

256 MB
Ulaz
Standardni ulaz
Izlaz
Standardni izlaz

Rimljani ponovo napadaju. Ovoga puta ih ima mnogo više od broja Gala u selu. Ali, Asterix se sprema za pobedu. Aleksandar Veliki je rekao: “Ne plašim se vojske lavova koju predvodi ovca, plašim se vojske ovaca koju predvodi lav.” (grčki: Μέγας Αλέξανδροςς, O Mégas Aléxandros, 356. pne. - 11. 6. 323. pne.),

Stoga računa koliko je slaba rimska vojska.

Asterix smatra da je slaba tačka neke vojske broj trojki i, j, k takvih da $i < j < k$ i $a_i > a_j > a_k$ gde je a_p snaga vojnika koji stoji na poziciji p. Rimska vojska ima jednu specijalnu osobinu – zna se tačna snaga svakog vojnika.

Pomozi Asterixu da izračuna koliko su slabi Rimljani.

Ulaz

U prvoj liniji se nalazi broj n ($3 \leq n \leq 10^6$) — broj vojnika u Rimskoj vojsci. Sledeća linija sadrži n različitih pozitivnih brojeva a_i ($1 \leq i \leq n$, $1 \leq a_i \leq 10^9$) — snagu svih vojnika rimske armije.

Izlaz

Jedan broj, slabost Rimljana.

Primeri

Primer1
Ulaz
```
3
3 2 1
```
Izlaz
```
1
```

Primer2
Ulaz
```
3
2 3 1
```
Izlaz
```
0
```

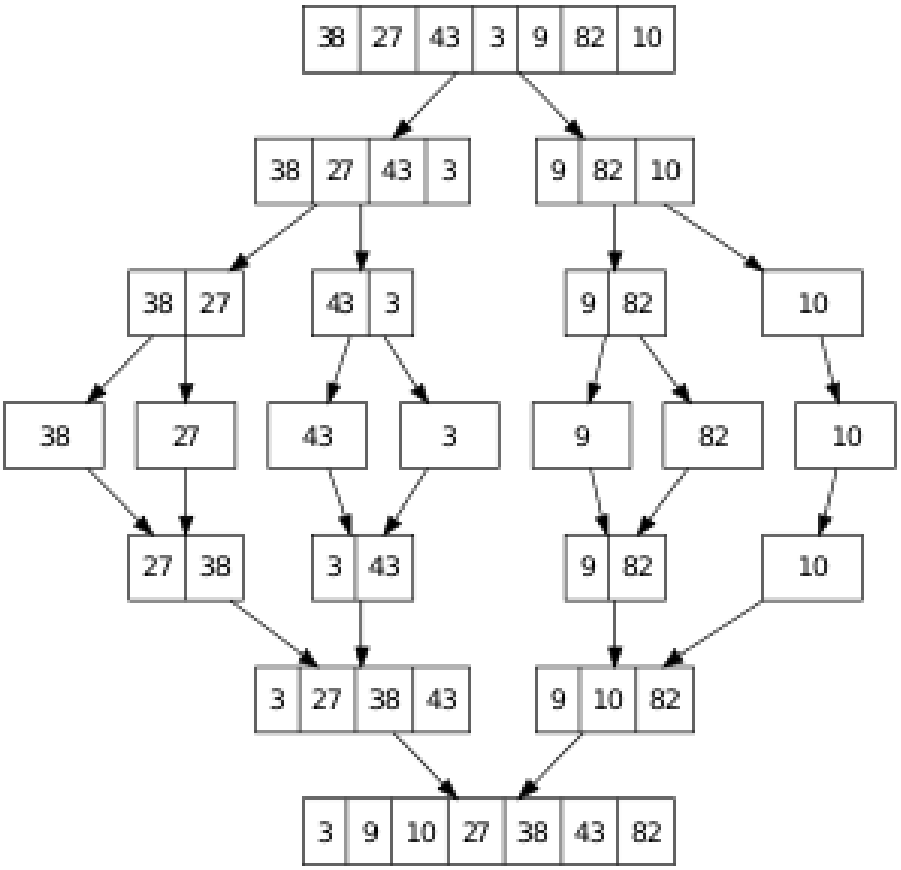
Primer3
Ulaz
```
4
10 8 3 1
```
Izlaz
```
4
```

Primer4
Ulaz
```
4
1 5 4 3
```
Izlaz
```
1
```

Pomoć

1. Segment tree

2. Merge sort




```
void merge_sort(int a[], int l, int d)
{
  int s;
  static int b[MAX];            /* Pomocni niz */
  int i, j, k;

  /* Izlaz iz rekurzije */
  if (l >= d)
    return;

  /* Odredjivanje sredisnjeg indeksa */
  s = (l + d) / 2;

  /* Rekurzivni pozivi */
  merge_sort(a, l, s);
```

```
  merge_sort(a, s + 1, d);

  /* Inicijalizacija indeksa. Indeks i prolazi krozi levu polovinu
     niza, dok indeks j prolazi kroz desnu polovinu niza. Indeks k
     prolazi kroz pomocni niz b[] */
  i = l;
  j = s + 1;
  k = 0;

  /* "Ucesljavanje" koriscenjem pomocnog niza b[] */
  while (i <= s && j <= d) {
    if (a[i] < a[j])
      b[k++] = a[i++];
    else
      b[k++] = a[j++];
  }

  /* U slucaju da se prethodna petlja zavrsila izlaskom promenljive
     j iz dopustenog opsega u pomocni niz se prepisuje ostatak leve
     polovine niza */
  while (i <= s)
    b[k++] = a[i++];

  /* U slucaju da se prethodna petlja zavrsila izlaskom promenljive
     i iz dopustenog opsega u pomocni niz se prepisuje ostatak desne
     polovine niza */
  while (j <= d)
    b[k++] = a[j++];

  /* Prepisuje se "ucesljani" niz u originalni niz */
  for (k = 0, i = l; i <= d; i++, k++)
    a[i] = b[k];
}
```

Ideja u zadatku 2:

Za svaki indeks ***i***, nađite broj svih inverzija u kojima je indeks ***i*** prvi član. Potom nađite broj svih inverzija u kojima je indeks ***i*** drugi član. Potom po pravilu proizvoda, pomnožimo ova dva broja da nađemo broj 3-inverzija trojki (j,i,k) gde indeks ***i*** je u sredini.