

Računarski praktikum 1, 2021 godina

Prva zadaća: postar, kuca

Imamo jedan grad u kojem poštari raznose poštu po kućama. Napišite sučelje i implementaciju za strukture `kuca` i `postar`.

Struktura `kuca` predstavlja neku kuću u nekom gradu. Svaka kuća ima koordinate koje predstavljaju lokaciju kuće unutar gradskih blokova (X i Y, oba `int`) i adresu na kojoj se nalazi (`string`).

Struktura `postar` predstavlja poštara. Svaki poštara ima maksimalnu količinu koraka (`int`), potrošnju (koraka) po jedinici udaljenosti (`int`) te kolekciju tipa `kuca` koji predstavljaju rutu koju poštara mora proći u jednom hodanju (najviše 100 `kuca`).

Sučelje (deklaraciju) za obje strukture spremite u datoteku `postar.h`, a implementaciju u datoteku `postar.cpp`.

Strukture moraju imati sljedeće elemente (smijete dodati i elemente po želji kako biste ostvarili sve funkcionalnosti):

Struktura `kuca` sadrži funkcije:

- `kuca(string adresa, int x, int y)` - Konstruktor, stvara kuću sa zadanim parametrima. Možete pretpostaviti da su koordinate uvijek pozitivne.
- `int udaljenost(kuca k)` - Vraća udaljenost do kuće `k`. Udaljenost računamo kao zbroj apsolutnih vrijednosti razlika po koordinatama: za točke (x_1, y_1) i (x_2, y_2) udaljenost je $|x_1 - x_2| + |y_1 - y_2|$.
- `string adresa()` - Vraća adresu na kojoj se kuća nalazi.

Struktura `postar` sadrži funkcije:

- `postar(int maxkoraka, int potrosnja)` - Konstruktor, stvara poštara sa zadanim parametrima. Broj `maxkoraka` označava koliko koraka poštara ima na raspolaganju. Broj `potrosnja` definira koliko je potrošnja koraka po jedinici udaljenosti, tj. broj koraka po jedinici udaljenosti. Stvoreni poštara ima praznu rutu (ne sadrži niti jednu kuću) te maksimalni broj koraka.
- `int brojkuca()` - Vraća trenutni broj kuća u ruti poštara.
- `void odmor()` - Postavlja preostali broj koraka na maksimalnu vrijednost.
- `int preostalokoraka()` - Vraća preostali broj koraka.
- `bool dodajkucu(kuca s, int k)` - Dodaje kuću u rutu poštara. Kuće se numeriraju od nule, tj. ako je `k` jednak nuli kuća se ubacuje na sam početak rute. Ako

poštar već prolazi 100 kuća ili k je manje od nule ili je kuća s istom adresom već u ruti, funkcija ne radi ništa i vraća `false`. U suprotnom, dodaje kuću na k -to mjesto u ruti poštara i vraća `true`. Ako je indeks k veći od trenutne duljine rute, kuća se dodaje na kraj, inače se ubacuje se na k -ti indeks, a kuće nakon k -tog indeksa se pomiču se za jedno mjesto u desno.

- `bool maknikucu(string adresa)` – Ukoliko takva postoji, iz rute miče kuću na danoj adresi i vraća `true`. Ako je kuća maknuta sve kuće nakon te kuće (tog indeksa kuće) se pomiču za jedno mjesto lijevo. Ukoliko nema takve kuće, ne radi ništa i vraća `false`.
- `kuca hoda()` – Poštar kreće od prve kuće u svojoj ruti i prolazi po svim kućama u ruti. Ukoliko poštar nema dovoljno koraka da stigne do sljedeće kuće, ne nastavlja dalje s hodaњem i vraća kuću na kojoj je stao. Inače, vraća zadnju kuću. Možete pretpostaviti da funkcija neće biti pozvana na poštaru s praznom rutom.

Primjer klijentskog programa

```
#include <iostream>
#include <string>
#include "postar.h"

using namespace std;

int main() {
    postar p1(20, 3);
    postar p2(10, 2);

    cout << p1.brojkuca() << endl; // 0
    cout << p2.brojkuca() << endl; // 0

    kuca k1("prva", 3, 3);
    kuca k2("druga", 4, 5);
    kuca k3("treca", 2, 5);
    kuca k4("cetvrta", 3, 4);
```

```

cout << k1.udaljenost(k2) << endl; // 3, jer  $|4-3|+|5-3| = 3$ 
cout << k2.udaljenost(k3) << endl; // 2
cout << k3.udaljenost(k4) << endl; // 2

cout << p1.dodajkucu(k2, 13) << endl; // 1
cout << p1.dodajkucu(k2, 0) << endl; // 0, vec postoji
cout << p1.dodajkucu(k1, 0) << endl; // 1
cout << p1.brojkuca() << endl; // 2

cout << p1.preostalogoraka() << endl; // 20
cout << p1.hodaj().adresa() << endl; // druga
cout << p1.preostalogoraka() << endl; //11 jer  $11=(20-udaljenost*potrosnja)$ 

cout << p1.hodaj().adresa() << endl; // druga
cout << p1.preostalogoraka() << endl; // 2

cout << p1.hodaj().adresa() << endl; // prva
cout << p1.preostalogoraka() << endl; // 2
p1.odmori();
cout << p1.preostalogoraka() << endl; // 20

p2.dodajkucu(k1, p2.brojkuca()); // dodajemo na kraj
p2.dodajkucu(k2, p2.brojkuca());
p2.dodajkucu(k3, p2.brojkuca());
p2.dodajkucu(k4, p2.brojkuca());

cout << p2.brojkuca() << endl; // 4
cout << p2.hodaj().adresa() << endl; // treca
cout << p2.maknikucu(k2.adresa()) << endl; // 1
cout << p2.maknikucu(k3.adresa()) << endl; // 1

```

```

cout << p2.brojkuca() << endl; // 2
cout << p2.maknikucu(k3.adresa()) << endl; // 0
cout << p2.brojkuca() << endl; // 2
cout << p1.brojkuca() << endl; // 2

return 0;
}

```

Opće napomene:

- Strukture, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija `main()`!
- Nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od main-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u `.h` niti u `.cpp` datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacijâ koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa, ali taj klijentski program ne šalžete!

Zadaću sastavila marija.barkidjija@math.hr