

1. Napraviti dvije pokazivačke varijable *ptr1* i *ptr2* koji pokazuju na varijable *num1* i *num2* tipa *float*. *num1* i *num2* predstavljaju stranice pravougaonika. Potrebno je zatražiti od korisnika unos za *num1* i *num2*, a zatim u programu ispisati dužinu dijagonale pravougaonika. **NAPOMENA:** dijagonala pravougaonika *c* sa zadatim stranicama *a* i *b* se računa sljedećom formulom

$$c = \sqrt{a^2 + b^2}$$

U ovom zadatku potrebno je koristiti pokazivačke varijable za kalkulaciju dijagonale pravougaonika. Pretpostavljamo da će korisnik unositi pozitivne brojeve tipa *float*.

2. Deklarisati strukturu *Tacka*, gdje ima dvije varijable članice tipa *double* nazvani *x* i *y*. U main funkciji deklarirati dvije tačke *tacka_1* i *tacka_2*. Pokretanjem programa, traži od korisnika unos za *tacka_1* i *tacka_2* u novim redovima redom *x1*, *y1*, *x2*, *y2* u novim redovima. Nakon unosa, program računa Manhattan distancu između datih tačaka. **NAPOMENA:** Manhattan distanca $d(a, b)$ je, zapravo, zbir apsolutnih razlika koordinata između tačaka *a* i *b*

$$d(a, b) = |a_x - b_x| + |a_y - b_y|$$

3. Deklarisati strukturu *Flasa*, koja ima sljedeće osobine sa tipovima: **volumen double** (označeno u litrima), **ambalaža string**, **proizvodjac string**, **recikliranje bool**. Zatim, napraviti funkciju *void ispisi_flasu(const Flasa& flasa)* koja ispisuje instancu *flasa*. Naprimjer, ako imamo *flasa* sa osobinama redom 2.0, *Plastika*, *Tesanjski Kiselj*, *false*, tada ispisujemo:

```
Proizvodjac: Tesanjski Kiselj  
Volumen: 2L  
Ambalaza: Plastika  
Da li se moze reciklirati: Ne
```

U main funkciji napisati testni program pri kojem ćemo imati tri instance tipa *Flasa* i za svaku instancu *Flasa* ispisuje na ekran sa praznim redom između.

4. Razvijamo sistem Diskont pića koji će se nazivati Twingo sistem. Iskoristit ćemo strukturu *Flasa* i dodati novu osobinu cijena *double*, koja daje cijenu instancama *Flasa* sa PDV-om. Definirati funkciju *double ukupna_cijena(Flasa* artikli, int broj_artikala)* koja će sabrati zbir cijena flaša u nizu *artikli*. Definirati funkciju *void unesi_flasu(Flasa& flasa)* koja će izvršiti unos flaše sa tastature. Zatim, u main funkciji napraviti program koji od korisnika traži prirodan broj *broj_artikala*, zatim za *broj_artikala* traži unos flaše, te na kraju ispisuje ukupnu cijenu artikala koristeći funkciju *ukupna_cijena*.
5. **(CRNA POGIBIJA)** Diskont pića Twingo odlučio je da uvede sistem recikliranja svojih flaša, pri čemu vraća 10% cijene flaše za recikliranje, ukoliko se može reciklirati. Svaki

kupac može donijeti svoje flaše da ubace u sistem reciklaže, međutim ne znači da svaka flaša se može reciklirati. Srećom, u našem dosadašnjem sistemu imamo osobinu *recikliranje* za naše flaše, pa ćemo to iskoristiti kao način za skeniranje flaša. Međutim, postoji problem: S obzirom da svaki kupac donosi svoje flaše, a ne možemo pretpostaviti da kupac zna sortirati svoje flaše, naš sistem reciklaže je dovoljno pametan da zna prepoznati flaše koje se mogu reciklirati, a koje ne.

Stoga, program funkcioniše na sljedeći način: Moramo napraviti još jednu osobinu flaše, a to je *barkod int*, koji je zapravo identifikacioni broj flaše. U vašem programu u prvom koraku definišete koliko vrsta flaša želite napraviti i napravimo niz mogućih flaša.

U sljedećem koraku, program traži sada od prvog korisnika unos barkodova za svoje flaše. Obratiti pažnju da se unese ispravan barkod. Tokom unosa, program u pozadini spašava barkod flaše u nekom nizu. Korisnik unosi sve dok ne unese neki broj manji od 0. Smatrat ćemo da je prvi korisnik završio sa unosom flaša i pritisnuo dugme za reciklažu. U pozadini programa flaše se dodaju u niz, jer je potrebno na kraju ispisati sve unesene flaše i koje flaše su uspješno reciklirane, te ispisuje za korisnika koliko će novca dobiti za reciklažu flaša. Program nastavlja dalje sa radom, očekujući unos od sljedećeg kupca.

Na ovaj način ćemo pomoći Diskontu pića da za svoj lager pića ima ispravan sistem reciklaže i kupci koji će dobiti svoju nagradu za recikliranje. **NAPOMENA:** Obratiti pažnju na veličinu niza tipa Flaše i povećavati ga po potrebi. Tipa, kapacitet niza flaša neka bude za početak 8, zatim kada dođemo do 9. artikla, potrebno je napraviti novi niz od 16 elemenata i prekopirati sadržaj niza u ovaj novi i obrisati stari niz. Ukoliko dođe više od 16 elemenata, potrebno je ponovo uduplati kapacitet, to jeste, novi niz od 32 elementa, kopirati stari sa 16 elemenata, obrisati i slično. Nije dozvoljeno koristiti std strukture tipa vector, deque i slično.