

Zadaća 3

Objektno orijentirano programiranje

Decembar, 2017

Sadržaj

1	Napomena	2
2	Problem 1	2
3	Problem 2	2
4	Problem 3	4
5	Problem 4	4
6	Problem 5	5

1 Napomena

Rok za završetak zadaće je 24.12.2017.

2 Problem 1

Napisati generičku funkciju `jednako` koja poredi sekvence `[od1, do1)` i `[od2, do2)`, pri čemu se pretpostavlja da sekvence imaju isti broj elemenata. Funkcija vraća tačnu vrijednost ukoliko su svi elementi dvaju sekvenci isti u suprotnom vraća netačnu vrijednost. Funkcija uzima tri parametra, prva dva označavaju početak i kraj prve sekvence, a treći parametar označava početak druge sekvence. Napomena: iteratori koji operiraju na sekvencama mogu biti različitog tipa.

Koristeći funkciju `jednako`, napisati program koji učitava riječi koje korisnik unosi sa tastature i ispisuje svaku riječ koju identifikira kao palindrom. Palindrom je riječ koja ima isto značenje bez obzira da li se čita sa lijeva na desno ili obrnuto. Primjer pozivanja programa:

```
$ ./p1
pop
pop: Palindrom
ako
svako
tako
ratar
ratar: Palindrom
bilo
ili
ili: Palindrom
^D
```

Opciono, napisati isti program koristeći algoritam `copy_if` definiran u zaglavlju `<algorithm>` i `stream` iteratore za unos i ispis.

3 Problem 2

Napisati program koji učitava rezultate utakmica Engleske lige od 2000 godine. Fajl sadrži informacije o sezoni, imenu domaćina, gosta, broju postignutih golova domaćina, kao i gosta, ligi i rezultatu, respektivno. Rezultat 'A' označava pobjedu gosta, 'H' pobjedu domaćina i 'D' neriješen rezultat. Nakon učitavanja korisniku omogućiti da odabere sezonu i ligu za ispis

tabele, sve dok korisnik eksplicitno ne prekine unos sa EOF uslovom. Za odabranu godinu(sezonu) i ligu potrebno je ispisati tabelu klubova sortiranu po pozicijama na tabeli, koja je određena ukupnim brojem bodova tima, tako što za svaku pobjedu tim dobija 3 boda, a za nerješen rezultat oba tima dobivaju po 1 bod. U slučaju da više timova ima isti broj bodova, pozicija se određuje na osnovu gol razlike (proj postignutih - broj primljenih golova). Pored toga, za svaki tim je potrebno ispisati broj pobjeda, poraza i nerješениh utakmica, kao i broj postignutih, primljenih golova i gol razliku.

U slučaju neispravnog unosa sezone ili lige, korisniku ispisati odgovarajuću poruku.

U prilogu je dat csv fajl `engleska_liga.csv` koji sadrži ranije navedene informacije o utakmicama. Podaci u jednom redu su razdvojeni sa `;`. Za rad sa fajlovima može se koristiti objekat tipa `std::ifstream` iz zaglavlja `<fstream>`. Objekat se može definisati na sljedeći način:

```
std::ifstream in("imefajla");
```

Dalje je moguće tretirati objekat `in` na identičan način kao i objekat `cin` npr

```
std::string x;
in >> x;
```

čita sljedeći string iz datoteke sa imenom `imefajla` i njegovu vrijednost dodjeljuje varijabli `x`. Objekat `in` je moguće i slati u funkcije koje očekuju `cin` objekat (tj. koje očekuju objekat tipa `std::istream`. Primjer pozivanja programa:

```
$ ./p2
```

Rezultati su učitani.

Unesite sezonu i ligu za prikaz tabele:

2013 1

POZ	KLUB	POB	NER	IZG	PGOL	PRGOL	GR	BOD
1	Manchester United	28	5	5	86	43	43	89
2	Manchester City	23	9	6	66	34	32	78
3	Chelsea	22	9	7	75	39	36	75
4	Arsenal	21	10	7	72	37	35	73
5	Tottenham Hotspur	21	9	8	66	46	20	72
6	Everton	16	15	7	55	40	15	63
7	Liverpool	16	13	9	71	43	28	61
8	West Bromwich Albion	14	7	17	53	57	-4	49

9	Swansea City	11	13	14	47	51	-4	46
10	West Ham United	12	10	16	45	53	-8	46
11	Norwich City	10	14	14	41	58	-17	44
12	Fulham	11	10	17	50	60	-10	43
13	Stoke City	9	15	14	34	45	-11	42
14	Southampton	9	14	15	49	60	-11	41
15	Aston Villa	10	11	17	47	69	-22	41
16	Newcastle United	11	8	19	45	68	-23	41
17	Sunderland	9	12	17	41	54	-13	39
18	Wigan Athletic	9	9	20	47	73	-26	36
19	Reading	6	10	22	43	73	-30	28
20	Queens Park Rangers	4	13	21	30	60	-30	25

Unesite sezonu i ligu za prikaz tabele:

~D

4 Problem 3

Implementirati generičku f-ju `akumuliraj` koja sumira elemente sekvence određene iteratorima `[pocetak, kraj)` sa nekom početnom vrijednošću `init`. Funkcija vraća rezultat koji je istog tipa kao i početna vrijednost. Elementi sekvence ne moraju biti istog tipa kao i početna vrijednost, ali moraju imati definiran `operator+`, te mora biti moguća konverzija iz tog tipa u tip početne vrijednosti. Funkcija prima tri parametra, `pocetak`, `kraj` i `init`. Napomena: funkcija `akumuliraj` je jedostavna implementacija algoritma `std::accumulate` definiranog u zaglavlju `<numeric>`. Napisati i odgovarajući glavni program kojim će se testirati navedena funkcija, za različite kombinacije ulaznih parametara.

5 Problem 4

Implementirati generičku f-ju `particionisi` koja elemente sekvence određene iteratorima `[pocetak, kraj)` regrupaše tako da svi elementi za koje unarna predikat funkcija `p` vrati `true` idu u prvi dio sekvence, a svi ostali elementi u drugi dio sekvece. Funkcija vraća iterator na prvi element drugog dijela sekvence ili iterator koji odgovara kraju sekvence, ukoliko svi elementi pripadaju prvom dijelu. Tip parametra predikat funkcije je takav da je moguća implicitna konverzija elemenata sekvence u taj tip, nakon dereferenciranja iteratora. Funkcija prima tri parametra, `pocetak`, `kraj` i `p`.

Napomena: funkcija `partition` je jednostavna implementacija algoritma `std::partition` definiranog u zaglavlju `<algorithm>`. Napisati i odgovarajući glavni program kojim će se testirati navedena funkcija, za različite kombinacije ulaznih parametara.

6 Problem 5

Implementirati klasu `BazaRadnika` koja implementira bazu radnika i omogućava:

- Unos pojedinačnog radnika u bazu. Baza ne dozvoljava unos dva radnika istog imena. Ukoliko se pokuša unijeti radnik dva puta metod generira `domain_error`
- Brisanje radnika određenog imena iz baze. Ako radnik nije izbrisan metod vraća `false`.
- Unos i ispis kompletne baze pomoću objekata klase `istream` i `ostream`
- Računanje prosječne plate i starosti radnika.

Klasu testirati sljedećim glavnim programom:

```
#include "radnik.h"
#include "baza.h"
#include <iostream>

int main()
{
    BazaRadnika b1;
    b1.ucitaj(std::cin);
    if (b1.izbrisi_radnika("alen"))
        std::cout << "alen izbrisan" << std::endl;
    std::cout << "U bazi su" << std::endl;
    b1.ispisi(std::cout);
    std::cout << "Prosjek plata je: " << b1.prosjek_plata() << std::endl;
    std::cout << "Prosjek godina je: " << b1.prosjek_godina() << std::endl;
    return 0;
}
```

Primjer pozivanja:

```
$ ./p5
selim 2300.4 34
alen 1950.6 55
alan 2130 59
alen 1200 23
Radnik vec u bazi
armando 5000 29
^D
alen izbrisan
U bazi su
alan 2130 59
armando 5000 29
selim 2300.4 34
Prosjek plata je: 3143.47
Prosjek godina je: 40.6667
```