*Domáce zadanie 11*

*Zadanie (na samostatnú prácu)*

Zdrojový kód riešenia zadania príkladu 11.1 rozšírte a upravte tak, aby program

vypisoval všetky prvky zoznamu, čiže údaje o všetkých načítaných študentoch

**usporiadane podľa dátumu narodenia študentov** od najstaršieho študenta po

najmladšieho študenta.

V zdrojovom kóde riešenia zadania príkladu 11.1 bude potrebné zmeniť a upraviť telo

funkcie

*int PorovnajS(TStudent a, TStudent b);*

a prípadne vykonať ďalšie úpravy v zdrojovom kóde riešenia zadania príkladu 11.1.

--------vypis zoznamu--------------

pocet studentov: 5

Z\_arku Johanka (C) 5. 4. 1238

Jankovic Jan (A) 4. 6. 1985

Jankovic Peter (A) 12. 6. 1985

Tell Ferdinand (C) 4. 12. 1986

Tell Viliam (F) 1. 1. 1987

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

podla coho chcete vyhladavat studenta? meno(0)/priezvisko(1): 1

zadajte retazec, ktory chcete vyhladat: tell

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

najdeni studenti:

Tell Ferdinand (C) 4. 12. 1986

Tell Viliam (F) 1. 1. 1987

pocet najdenych studentov: 2

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

#include<fstream>

#include<string.h>

#include<iostream>

using namespace std;

//-----------------datove struktury-------------------//

struct datum

{

int d, m, y;

};

struct TStudent

{

char meno[32], priezvisko[32];

datum datum\_narodenia;

char znamka\_M;

};

struct TPrvok

{

TStudent student;

TPrvok \*dalsi;

};

struct TZoznam

{

TPrvok \*prvy, \*posledny;

};

//-------------------funkcie-----------------------------//

int JePrazdny(TZoznam z)

{

return z.prvy == NULL;

}

int PorovnajS(TStudent a, TStudent b)

{ // podla priezviska, mena, datumu narodenia

long da, db;

da = a.datum\_narodenia.d + a.datum\_narodenia.m \* 100 + a.datum\_narodenia.y \* 10000;

db = b.datum\_narodenia.d + b.datum\_narodenia.m \* 100 + b.datum\_narodenia.y \* 10000;

if (da > db) return 1;

if (da < db) return -1;

return 0;

}

void Vloz(TZoznam &z, TStudent x) // vlozi 1 prvok usporiadane do zoznamu

{

TPrvok \*novy = new TPrvok;

novy->student = x;

novy->dalsi = NULL;

if (z.prvy == NULL) // prazdny zoznam

z.prvy = z.posledny = novy;

else if (PorovnajS(x, z.prvy->student) == -1) // (x < z.prvy->x) studenta treba vlozit na zaciatok zozn.

{

novy->dalsi = z.prvy;

z.prvy = novy;

}

else if (PorovnajS(x, z.posledny->student) == 1) // (x > z.posledny->x) studenta treba vlozit na koniec

// zoznamu

{

z.posledny->dalsi = novy;

z.posledny = novy;

}

else // inak studenta treba vlozit niekde medzi existujuce prvky zoznamu; v cykle while hladame,

{ // medzi ktore prvky ho treba vlozit

TPrvok \*p1 = z.prvy, \*p2 = z.prvy->dalsi;

while (PorovnajS(p2->student, x) == -1) // (p2->x < x)

{

p1 = p2; p2 = p2->dalsi;

}

p1->dalsi = novy;

novy->dalsi = p2;

}

}

TPrvok\* Najdi(TZoznam z, char \*hladane, TPrvok \*start = NULL, int podla\_p = 1)

{ // ak parameter 'start' obsahuje inu hodnotu ako NULL, tak funkcia bude hladat od prvku, na ktory

// ukazuje tento ukazovatel dalej, inak bude hladat od zaciatku zoznamu

TPrvok \*p;

if (start == NULL)

p = z.prvy;

else

p = start->dalsi;

char hladane\_pr[32];

while (p != NULL)

{

if (podla\_p)

strcpy(hladane\_pr, p->student.priezvisko); // aby sme si nezmenili povodne priezvisko

else

strcpy(hladane\_pr, p->student.meno); // aby sme si nezmenili povodne meno

if (strcmp(\_strlwr(hladane\_pr), \_strlwr(hladane)) == 0)

return p;

p = p->dalsi;

}

return NULL;

}

void VypisPrvok(TPrvok \*px)

{

cout << px->student.priezvisko << " " << px->student.meno << " \t(" << px->student.znamka\_M << ") ";

cout << px->student.datum\_narodenia.d << ". " << px->student.datum\_narodenia.m << ". ";

cout << px->student.datum\_narodenia.y << endl;

}

void Vypis(TZoznam z)

{

if (JePrazdny(z)) return;

TPrvok \*p = z.prvy;

while (p->dalsi != NULL)

{

VypisPrvok(p);

p = p->dalsi;

}

VypisPrvok(p);

}

void ZmazPosledny(TZoznam &z)

{

if (z.prvy == NULL) // prazdny zoznam

return;

TPrvok \*p = z.prvy;

if (p->dalsi == NULL) // je len jeden prvok v zozname

{

delete p;

z.prvy = z.posledny = NULL;

return;

}

// najdeme predposledny prvok

while (p->dalsi->dalsi != NULL)

p = p->dalsi;

delete p->dalsi;

p->dalsi = NULL;

z.posledny = p;

}

void Zmaz(TZoznam &z)

{

while (!JePrazdny(z))

ZmazPosledny(z);

}

void Nacitaj(TStudent &s, istream &vstup)

{

vstup >> s.meno >> s.priezvisko >> s.znamka\_M;

vstup >> s.datum\_narodenia.d >> s.datum\_narodenia.m >> s.datum\_narodenia.y;

}

int main()

{

TZoznam zoznam;

zoznam.prvy = zoznam.posledny = NULL;

TStudent s;

int n = 0;

ifstream vstup;

vstup.open("prvaci.txt");

if (!vstup)

{

cout << "Nepodarilo sa otvorit vstupny subor";

return 0;

}

Nacitaj(s, vstup);

Vloz(zoznam, s);

while (s.datum\_narodenia.d)

{

Nacitaj(s, vstup);

if (s.datum\_narodenia.d)

Vloz(zoznam, s);

n++;

}

vstup.close();

cout << "--------vypis zoznamu--------------" << endl;

cout << "pocet studentov: " << n << endl << endl;

Vypis(zoznam);

cout << "-----------------------------------" << endl;

int kriterium;

cout << "\npodla coho chcete vyhladavat studenta? meno(0)/priezvisko(1): ";

cin >> kriterium;

cout << endl << "zadajte retazec, ktory chcete vyhladat: ";

char przv[32];

cin >> przv;

cout << "\n-----------------------------------\nnajdeni studenti:\n" << endl;

TPrvok \*najdeny = Najdi(zoznam, przv, NULL, kriterium); // hladame prvy vyskyt hladaneho studenta

int pocet = 0;

while (najdeny)

{

pocet++;

if (najdeny)

VypisPrvok(najdeny);

najdeny = Najdi(zoznam, przv, najdeny, kriterium); // hladame dalsie vyskyty hladaneho

// studenta od posledneho najdeneho studenta

}

cout << "\npocet najdenych studentov: " << pocet << endl;

cout << "-----------------------------------" << endl;

Zmaz(zoznam);

delete najdeny;

return 0;

}