**РУСЕНСКИ УНИВЕРСИТЕТ „АНГЕЛ КЪНЧЕВ“**

Катедра: Компютърни системи и технологии

**КУРСОВА ЗАДАЧА**

**по дисциплината**

**„Синтез и анализ на алгоритми“**

Изготвил: Алекс Георгиев Иванов

Фак. Номер: 233107

Група: 6А

Курс: 2

Специалност: КСТ

Дата: 11.12.2024 Приел:......................

/доц. д-р Е. Големанова/

1. **Задание**

**№13:** Общината в Русе се нуждае от софтуер за обработване на адресите жителите й. За да се изпълни желанието на общината е нужна програма за запазва на адресите и броя на жителите в едносвързан списьк. Всеки елемент от спись съдържа адрес и броя на всички хора, които живеят на този адрес.

Да се състави програма съдьржаща следните функции:

* функция за добавяне на нов адрес
* функция за добавяне на нов жител на общината, живеещ на зададен адрес
* функция за извеждане на броя на всички жители на общината

1. **Графично изображение на динамичния списък с конкретни примерни стойности**

First

Last

Null

Плиска 91 (адрес)

Сърнена Гора 5

next

3

4 (жители)

**4**

1. **Основни компоненти**
   1. Структура Node с полета:
      1. string address – адресът
      2. int residents – броя на жителите
      3. Node\* next – указател към следващият елемент в списъка
   2. Функции:
      1. void addAddress(Node\*& head, const string& address)
         1. Добавя нов адрес в списъка
         2. Ако списъка е празен, новият елемент става начало на списъка
         3. Ако не е празен, новият адрес се добавя в края на списъка
      2. void addResident(Node\* head, const string& address)
         1. Търси адреса в списъка
         2. Ако адреса бъде намерен увеличава броя на жителите на този адрес с 1
         3. Ако не бъде намерен връща съобщение за грешка
      3. void displayTotalResidents(Node\* head)
         1. Обхожда списъка и сумира броя на жителите за всички адреси
         2. Извежда общият брой жители
2. **Описание на използваните функции**

* void addAddress(Node\*& head, const string& address)
  + **Описание**: Добавя нов адрес в общността. Адресът се добавя в края на списъка.
  + **Параметри**:
    - Node\* head
    - const string& address: Адресът, който искаме да добавим.
  + **Действие**: Извежда съобщение, че нов адрес е добавен.
* **void addResident(Node\* head, const string& resident)**
  + **Описание**: Добавя нов жител към съществуващ адрес в списъка.
  + **Параметри**:
    - Node\* head
    - const string& address: Адресът, който искаме да добавим.
  + **Действие**: Преглежда всеки адрес в списъка. Ако адресът е намерен, добавя жителя към съответния списък от жители и извежда съобщение за успешно добавяне. Ако адресът не е намерен, извежда съобщение за грешка.
* **void displayTotalResidents(Node\* head)**
  + **Описание**: Връща общия брой на жителите в общността.
  + **Параметри**:
    - Node\* head
  + **Действие**: Обхожда всички адреси в списъка и сумира броя на жителите и го извежда в конзолата.

1. **Тестови примери (снимки на екрана), доказващи коректността на програмата**

**Описание**: Примерен код, който демонстрира създаването на общност, добавяне на адреси и жители, и извеждане на общия брой жители.

* 1. **Операции**:
     1. Добавя се адрес „ул. Плиска 91”.

A screenshot of a computer

Description automatically generated

* + 1. Добавя се адред „ул. Сърнена Гора 5“

A black screen with white text

Description automatically generated

* + 1. Добавят се четирима жители на адрес „Плиска 91“.

A screenshot of a computer

Description automatically generated

* + 1. Добавят се трима жители на адрес „Сърнена Гора 5“

A black screen with white text

Description automatically generated

* + 1. Добавя се един жител на адрес „Липсващ адрес“

A black background with white text

Description automatically generated

* + 1. Извежда се общият брой на жителите.

A black screen with white text

Description automatically generated

* + 1. Изход от програмата

A black background with white text

Description automatically generated

1. **Разпечатка на кода**

#include <iostream>

#include <cstring>

using namespace std;

// Структура за един елемент в списъка

struct Node

{

string address; // Адрес

int residents; // Брой жители

Node \*next; // Указател към следващия елемент

};

// Функции за управление на динамичния списък

void addAddress(Node \*&head, const string &address); // Добавяне на нов адрес

void addResident(Node \*head, const string &address); // Добавяне на жител към адрес

void displayTotalResidents(Node \*head); // Показване на общия брой жители

int main()

{

Node \*head = nullptr; // Инициализиране на началото на списъка

int choice;

string address;

// Основно меню

while (true)

{

cout << "\nМеню:\n";

cout << "1. Добавяне на нов адрес\n";

cout << "2. Добавяне на жител към адрес\n";

cout << "3. Показване на общия брой жители\n";

cout << "4. Изход\n";

cout << "Изберете опция: ";

cin >> choice;

switch (choice)

{

case 1:

cout << "Въведете адрес: ";

cin.ignore(); // Игнориране на символа за нов ред

getline(cin, address); // Четене на адрес

addAddress(head, address);

break;

case 2:

cout << "Въведете адрес за добавяне на жител: ";

cin.ignore(); // Игнориране на символа за нов ред

getline(cin, address); // Четене на адрес

addResident(head, address);

break;

case 3:

displayTotalResidents(head);

break;

case 4:

cout << "Изход от програмата.\n";

return 0; // Излизане от програмата

default:

cout << "Невалидна опция. Опитайте отново.\n";

}

}

return 0;

}

// Добавяне на нов адрес в списъка

void addAddress(Node \*&head, const string &address)

{

Node \*newNode = new Node; // Създаване на нов елемент

newNode->address = address; // Задаване на адрес

newNode->residents = 0; // Брой жители по подразбиране е 0

newNode->next = nullptr; // Следващият елемент е nullptr

// Добавяне на елемента в списъка

if (head == nullptr)

{

head = newNode; // Ако списъкът е празен, новият елемент става начало

}

else

{

Node \*temp = head;

while (temp->next != nullptr)

{ // Намиране на последния елемент

temp = temp->next;

}

temp->next = newNode; // Присвояване на новия елемент като последен

}

cout << "Адресът беше добавен успешно.\n";

}

// Добавяне на жител към конкретен адрес

void addResident(Node \*head, const string &address)

{

Node \*temp = head;

while (temp != nullptr)

{ // Обхождане на списъка

if (temp->address == address)

{ // Проверка дали адресът съвпада

temp->residents++; // Увеличаване на броя жители

cout << "Жителят беше добавен успешно.\n";

return;

}

temp = temp->next; // Преминаване към следващия елемент

}

cout << "Адресът не беше намерен.\n";

}

// Показване на общия брой жители

void displayTotalResidents(Node \*head)

{

int total = 0; // Общ брой жители

Node \*temp = head;

while (temp != nullptr)

{ // Обхождане на списъка

total += temp->residents; // Добавяне на броя жители за всеки адрес

temp = temp->next; // Преминаване към следващия елемент

}

cout << "Общ брой жители: " << total << endl;

}