Факултет по компютърни системи и технологии

ПИК 2

**Курсова работа**

Изготвила :

**Евелина Стефанова Кисьова**

Специалност : КСИ

Група: 49

Фак.ном.:121218089

**Документация**

* **Int main ()**  - Създаваме листа с данни от бинарния файл. Създаваме безкраен цикъл за отваряне на опциите .
* **NODE\* create\_list(void)** - Тази функция създава листа, като прочита информацията от бинарен файл и я зарежда в динамична структура . Връща указател към "корена"(върха) на листа, и не приема параметри .
* **NODE\* create\_node(NODE\* root, NODE temp) -** Тази функция създава нова структура и я добавя на върха на листа като заделя памет за един елемент структура , записва подадената информация като параметри в структурата и прави елемента да е новият връх . Връща указател към "корена"(върха) на листа и приема като параметри: старият "корен", и информацията за структурата .
* **void printList(NODE\* root)** - Функцията принтира на екрана целия лист като го обхожда и приема като параметър указателя към корена на листа .
* **void printEntry(NODE\* temp)**  - Функцията принтира елемент от листа на екрана и приема указател към съответния елемент .
* **void change\_entry(NODE\* root) -** Функцията прочита от потребителя, кой елемент иска да промени като прочита началната и крайната точка , след което обхожда целия лист и ако намери конкретния маршрут се принтира елемента , след което прочита новото времетраене . Приема като параметър указателя към корена на листа .
* **NODE\* delete\_node(NODE\* root)** - Функцията прочита от потребителя, кой елемент иска да изтрие и прочита началната и крайната точка , след което обхожда целия лист, като запазва кой е и предишния елемент , ако намери елемент, който отговаря , проверява дали иска да изтрие върха , запазва елемента след върха и освобождава върха. Свързва елементите, които са от двете страни на елемента, който иска да махне и освобожда елемента , след което връща връха . Приема като параметър указателя към корена на листа и връща указател към "корена"(върха) на листа .
* **NODE \* add\_new(NODE\* root)** – Функцията прочита началната и крайната точка на новия запис ,след което обхожда целия лист за да провери дали няма вече такъв елемент , ако има, прекратява функцията и връща върха на листа . Ако няма прочита информацията за новия елемент , проверя дали е валидна информацията , извиква функцията за създаване на нова структура, която връща указател към нов "корен" (връх) на листа и приема като параметри: старият "корен", и информацията за структурата .
* **void findFastestRoute(NODE\* root)** - Функцията търси най-малка стойност като почва от най-голямата , запазва най-бързия маршрут , обхожда целия лист и търси маршрута с най-малкото времетраене , след което принтира резултата . Приема като параметър указателя към корена на лист .
* **void findShortestRoute(NODE\* root)** – Функцията търси най-малка стойност като почва от най-голямата , запазва най-краткия маршрут , обхожда целия лист и търси най-кратката дистанция и принтира резулатата . Приема като параметър указателя към корена на листа .
* **void exitProgramm(NODE\* root)** – Функцията запазва на ново файла, като презаписва информацията ако има някакви промени. Приема като параметър указателя към корена на листа .