**Курсова работа по дисциплина ПИК2 за студенти ФКСТ**

**Емилиян Емилов Ризов - 121219091, 31гр, КСИ, ФКСТ**

**Десислав Андреев –**

**Описание на паметта спрямо отделните сегменти**

1. Code(Text) segment – 3.9KB
2. Global variables segment – 0b (Нямам глобални променливи)
3. Stack –

В зависимост от функцията, която се изпълнява

* 24bytes (main) + 68bytes(get\_info)
* 24bytes (main) + 76bytes(create\_list)
* 24bytes (main) + 40bytes (traverse)
* 24bytes (main) + 92bytes(equalize)
* 24bytes (main) + 60bytes(kpp)

1. Heap –

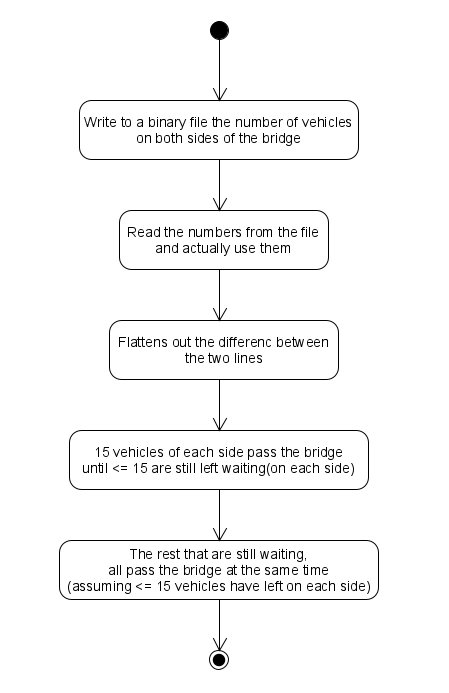
В зависимост от желаната дължина на списъка

* left\_number\*8bytes(размерът на struct l\_Vehicles)
* right\_number\*8bytes(размерът на struct r\_Vehicles)

**Описание на условието на задачата && диаграма на процесите, които представлява то**

Станала поледица на Дунав мост, както и авария в системата за обслужване на Гранична полиция. Моята цел е да напиша програма, която да пуска честен и за двете страни брой превозни средства(опашките да намаляват колкото се може по – равномерно), като се спазват условията

* не повече от 30 превозни средства на моста общо и от двете страни
* не повече от 20 превозни средства в една посока



**Файлова структура спрямо разделното компилиране**

Програмата се състои от:

* един главен сорс файл(finalApp.c)
* второстепенния главен сорс файл(main.c)
* отделен .с файл, който служи за инициализация на всички функции
* header file – служи за декларация на прототипите на функциите, както и за декларация и инициализация на двете структури

**Описание на структурите и техните роли**

Използваните структури са две

1. struct l\_Vehicles – съдържа два члена

* int val – служи като номер на съответното превозно средство или като място в колоната
* struct l\_Vehicles \*next – служи за създаване на връзка между отделните от свързания списък

1. struct r\_Vehicles – съдържа два члена

* int val – служи като номер на съответното превозно средство или като място в колоната
* struct r\_Vehicles \*next – служи за създаване на връзка между отделните от свързания списък

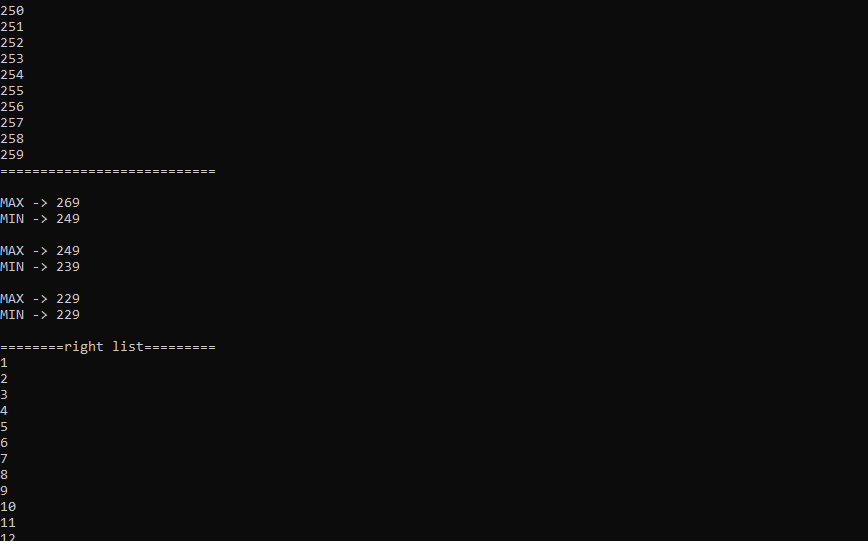
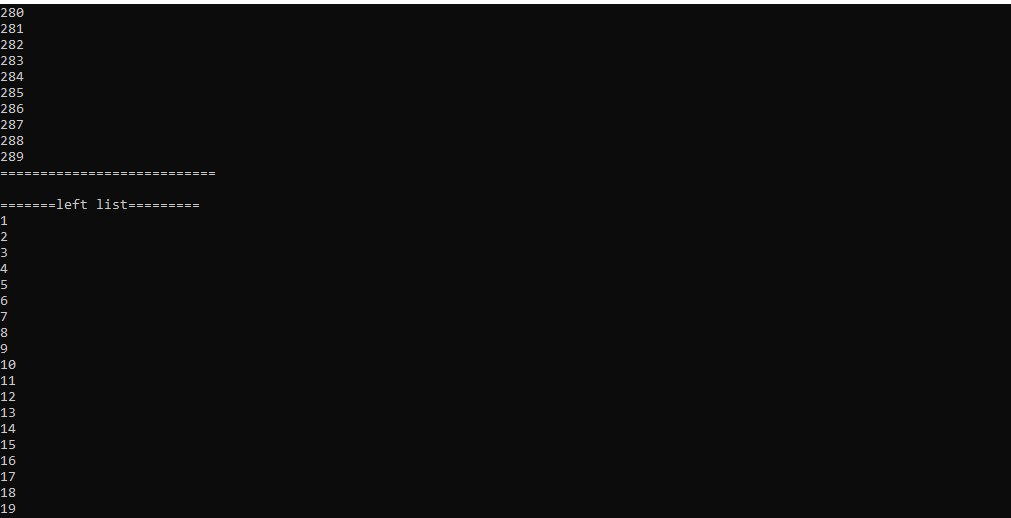
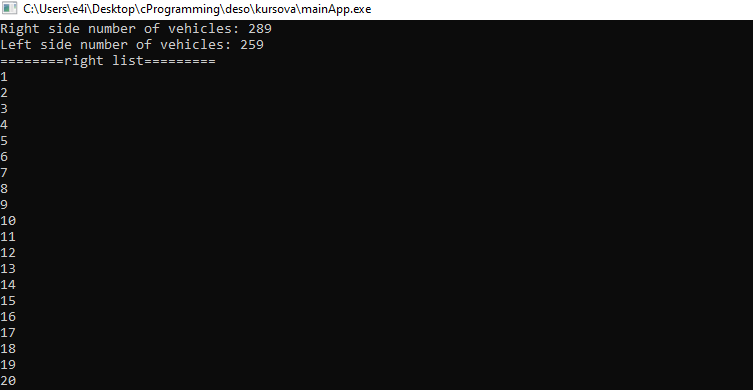
Главната и единствената им роля е да осъществяват връзка между всяко следващо превозно средство, представени като свързан списък.

**Описание на динамично заделената памет**

В програмата, динамична памет се ползва само при създаването на двата свързани списъци. По – точно при създаването на всеки един нов възел(node). Използвам malloc(), като за размер на заделената памет ползвам размерa, съответно на struct l\_Vehicles и на struct r\_Vehicles.

**Описание на логиката на програмата чрез УМЛ диаграми**

**Screenshot примери**

****