**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ ВАРНА**

**Факултет по изчислителна техника и автоматизация**

**Софтуерни и интернет технологии**

**семестриална домашна работа: факултет**

**вариант 109**

**Изготвил: Кристиян Юлиянов Бояров**

**21621600**

Вариант | **109**

**21621600 КРИСТИЯН ЮЛИЯНОВ БОЯРОВ**

**Агенция за имоти**

Да се напише компютърна програма, реализираща информационна система, която поддържа агенция за имоти (регистрационен номер на агенцията, списък с имоти). Програмата съхранява и обработва данни за имотите в агенцията. Максималният брой имоти (квартал, улица, Площ, Цена, брой стаи) в агенцията е 50.

# Базова задача – сложност ниска

1. Меню за избор на функциите в програмата. (7 седмица)
2. Добавяне на имоти в агенцията (7-8 седмица)
   1. Добавяне на един имот с координати в града (в града има 5 квартала (Младост,

Чайка, Левски, Център, Гръцки));

* 1. Добавяне на списък с имоти. Въвежда се цяло число ***n*** и след него ***n*** на брой имота.

1. Извеждане на всички имоти на екрана (8-9 седмица)
   1. Извеждане на имоти с максимална цена
   2. Извеждане на имоти в определен квартал

# Допълнение първо – сложност средна (+ базова задача)

D. Редактиране на имотите (10-11 седмица)

1. Въвежда се улица на имот, ако съществува имот на тази улица, се редактира цената му и лицето за контакт;
2. Имот се закупува по улица, когато се купи имот той се премахва от списъка с имоти.
3. Ако не съществува такъв имот, се извежда съобщение.

# Допълнение второ – сложност висока (+ базова задача + допълнение първо)

E. Търсене на имот: (9-10 седмица)

1. Извеждане на имоти по въведен квартал.
2. Ако не съществуват имоти във въведения квартал, се извеждат имоти от найблизките съседни квартали.

# Допълнение трето – сложност висока (+ базова задача + допълнение първо + допълнение второ)

1. Справки за имотите с под меню (11-12 седмица)
   1. Извеждане на всички имоти в подреден ред по:

i. Цена на имота; ii. Брой стаи.

* 1. Извеждане на имота с най-голяма квадратура;
  2. Извеждане на най-евтиния имот по цена за квадрат.

1. Данните в програмата да могат да се запазват във файл между две стартирания на програмата.

# Допълнение четвърто – (за допълнителни точки)

H. Допълнителни условия:

1. За точка B при добавянето на нов имот да се даде възможност за избор дали да се въведе нов или да се прекъсне въвеждането.
2. За точка C да се реализира извеждането на имоти по 5 на страница. Пример: Има въведени 50 имота. Извеждат се 5 имота, избира се следваща страница извеждат се следващите 5 и т.н.
3. За точка G да се записват данните в двоичен файл.

|  |  |  |
| --- | --- | --- |
| Чайка | Ц    Е    Н    Т    Ъ    Р | Гръцка |
| Левски | Младост |

**Трябва да се добавят тези файлове за да работи.**

За изпълнението на

* Базова задача са използвани - struct imot, int main, imot input, int main, int entryS, void append, void list, void max, void izvej\_kvartal, imot input
* Допълнение първо са използвани - struct imot, int main, void redaktirane\_cena, void zakupuvane
* Допълнение второ са използвани - struct imot, int main, void search\_kvartal,
* Допълнение трето са използвани - struct imot, int main, int spravka\_menu, void spravka\_cena, void spravka\_stai, void spravka\_kvadratura, void spravka\_cena\_kv, void loadfile, void filesave

struct imot {

char kvartal; -квартал на имота

char ulica[30]; -улица на имота

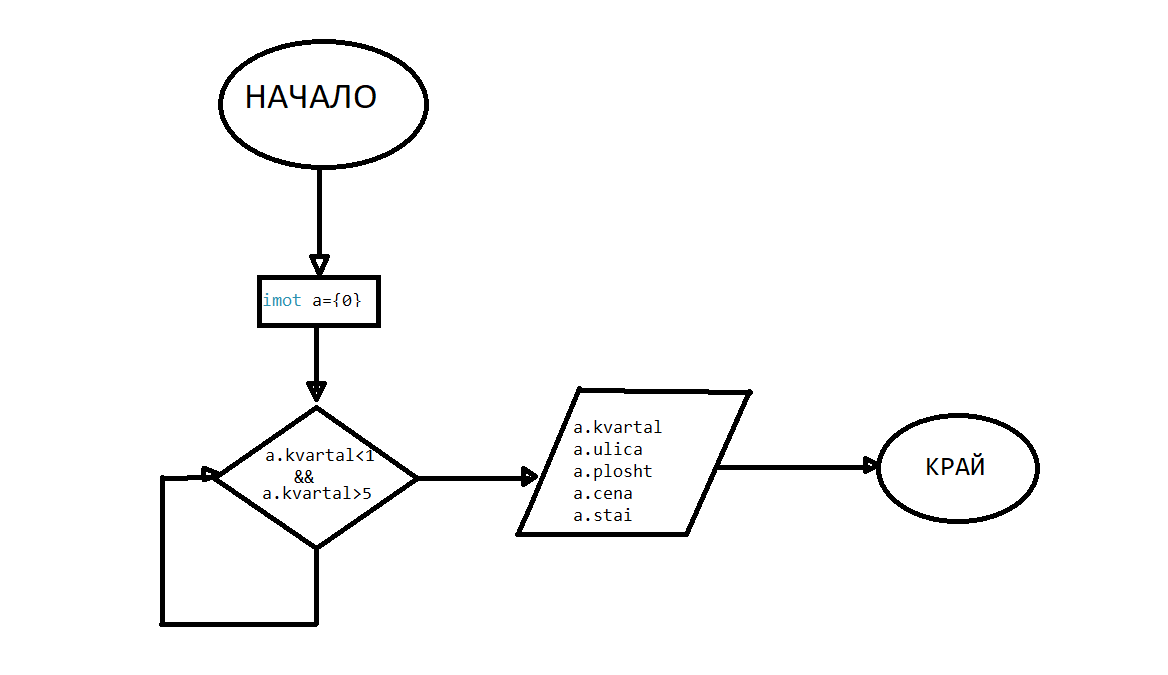
int plosht; -площ на имота

int cena; -цена

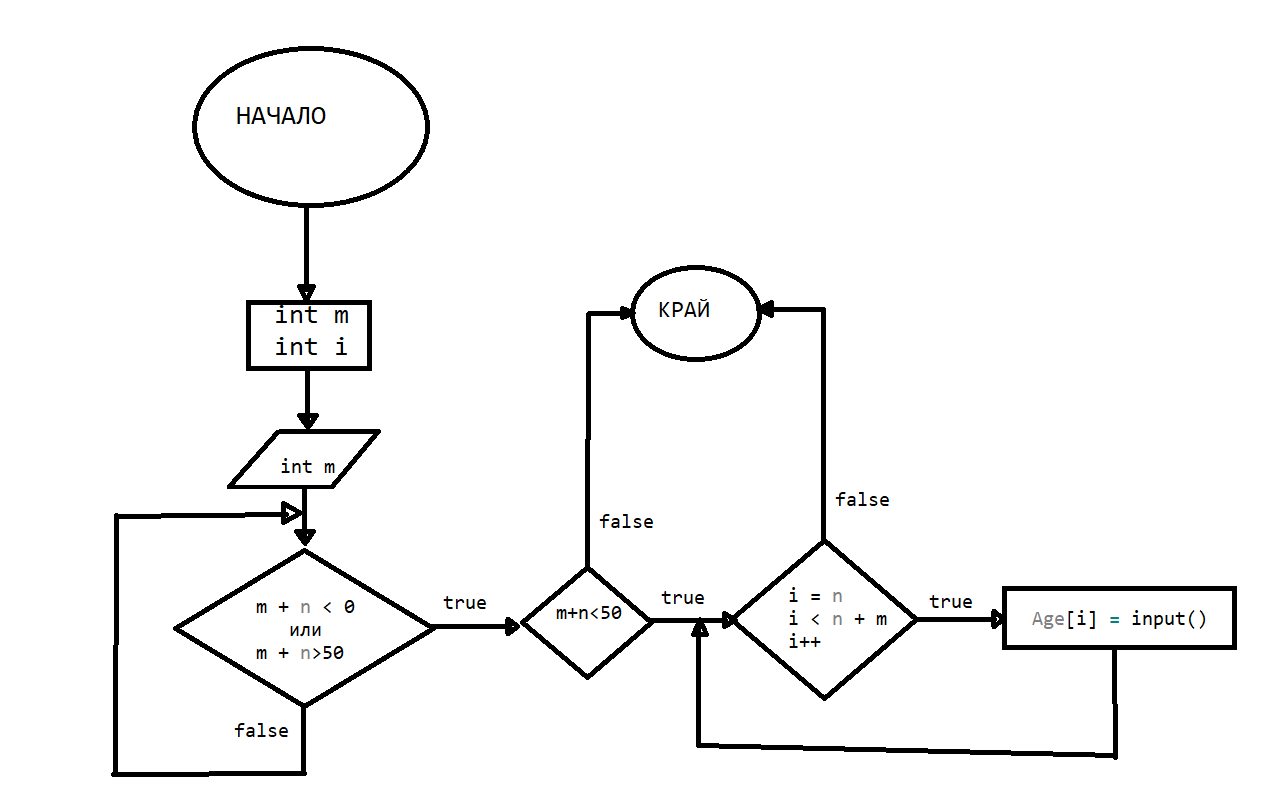
int stai; -стаи

};

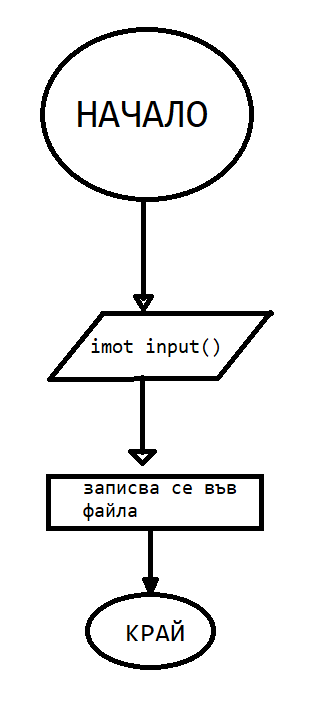
imot input() – въвежда информацията за имотите (квартал, улица, Площ, Цена, брой стаи)



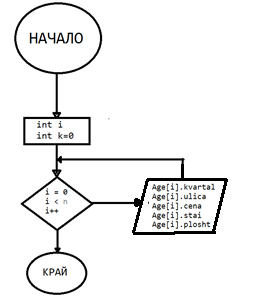
int entryS(imot Age[],int n)- служи за първоначалното въвеждане на няколко имота, въвежда се n на броя имоти, след което се въвежда информацията за имотите (квартал, улица, Площ, Цена, брой стаи)



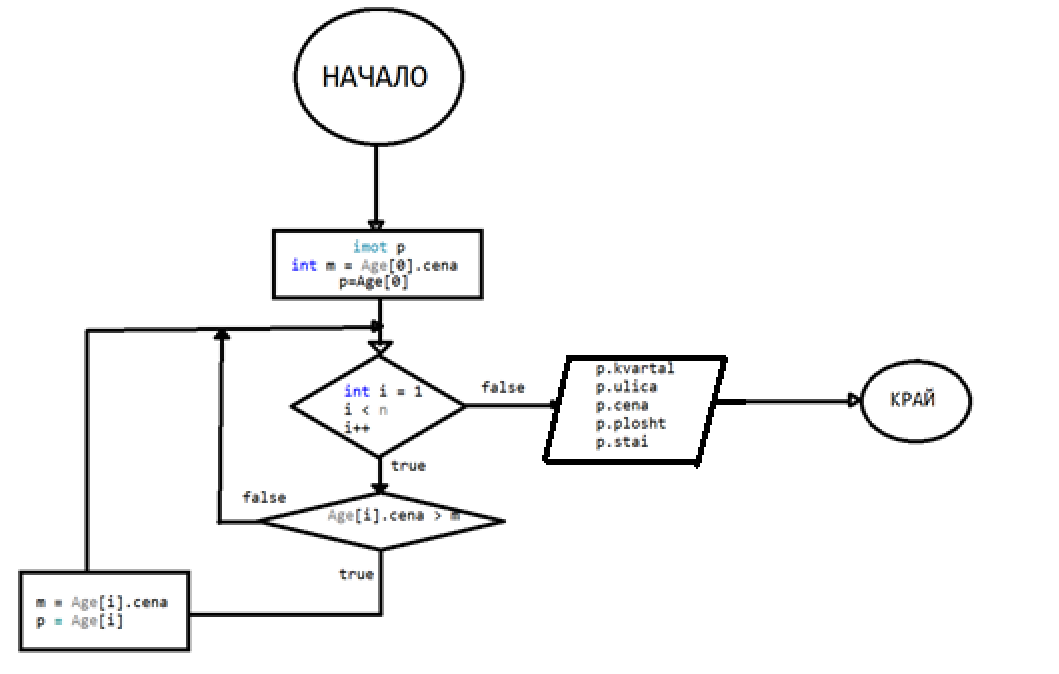
void append()-добавя само един имот, след което се запитва за добавянето на още един



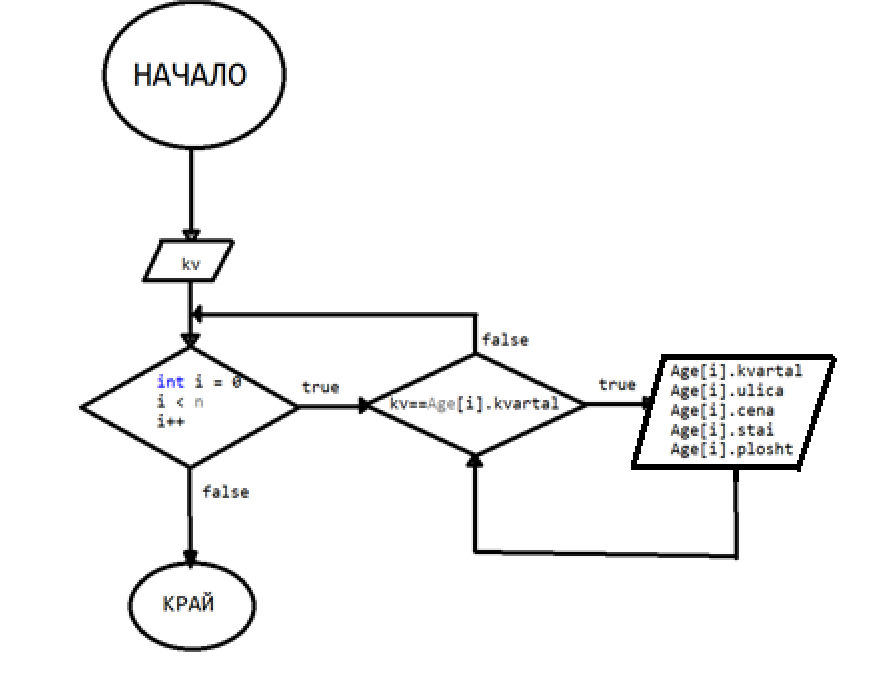
void list(imot Age[], int n)- извежда всички записани имоти



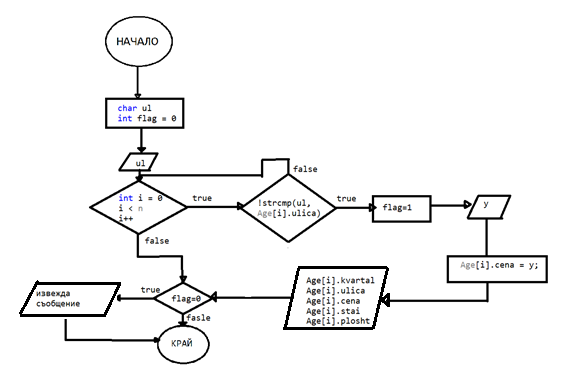
void max(imot Age[], int n)- от записаните имоти се извежда имота с най-висока цена



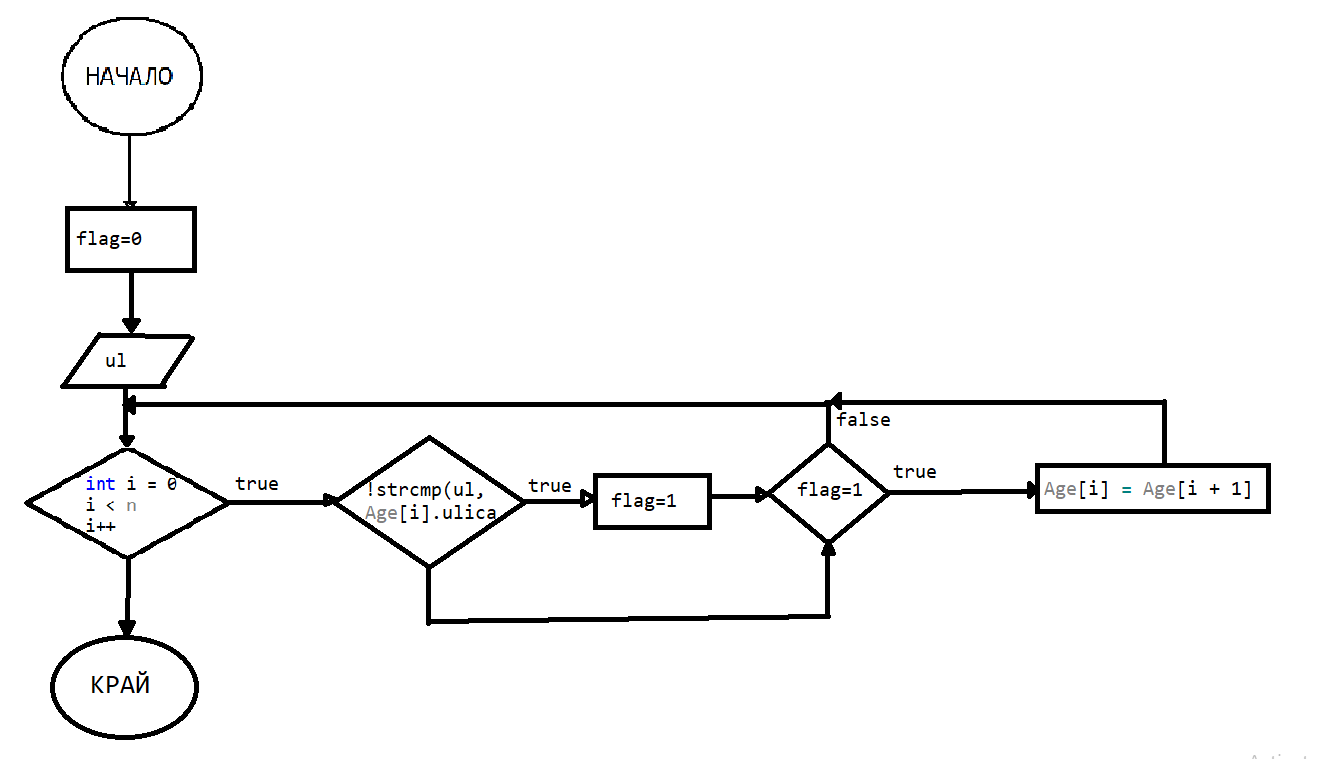
void izvej\_kvartal(imot Age[], int n)- записаните имоти се извеждат по зададен квартал



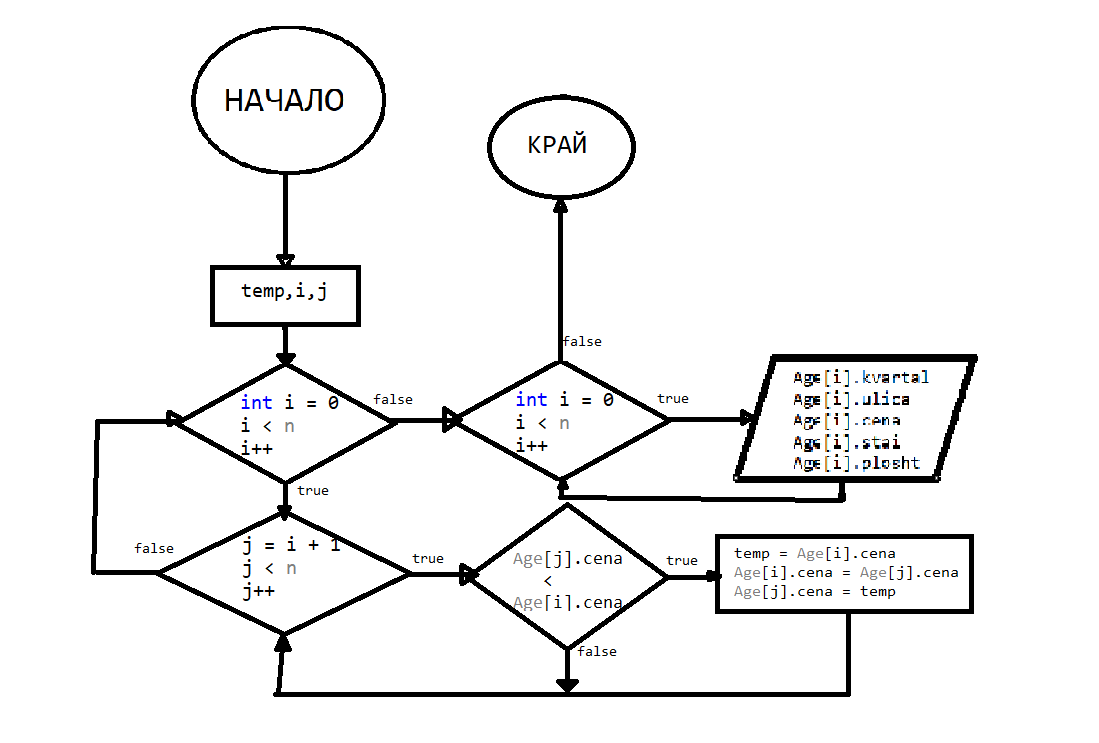
void redaktirane\_cena(imot Age[], int n)-търси имот по улица, след което редактира цената



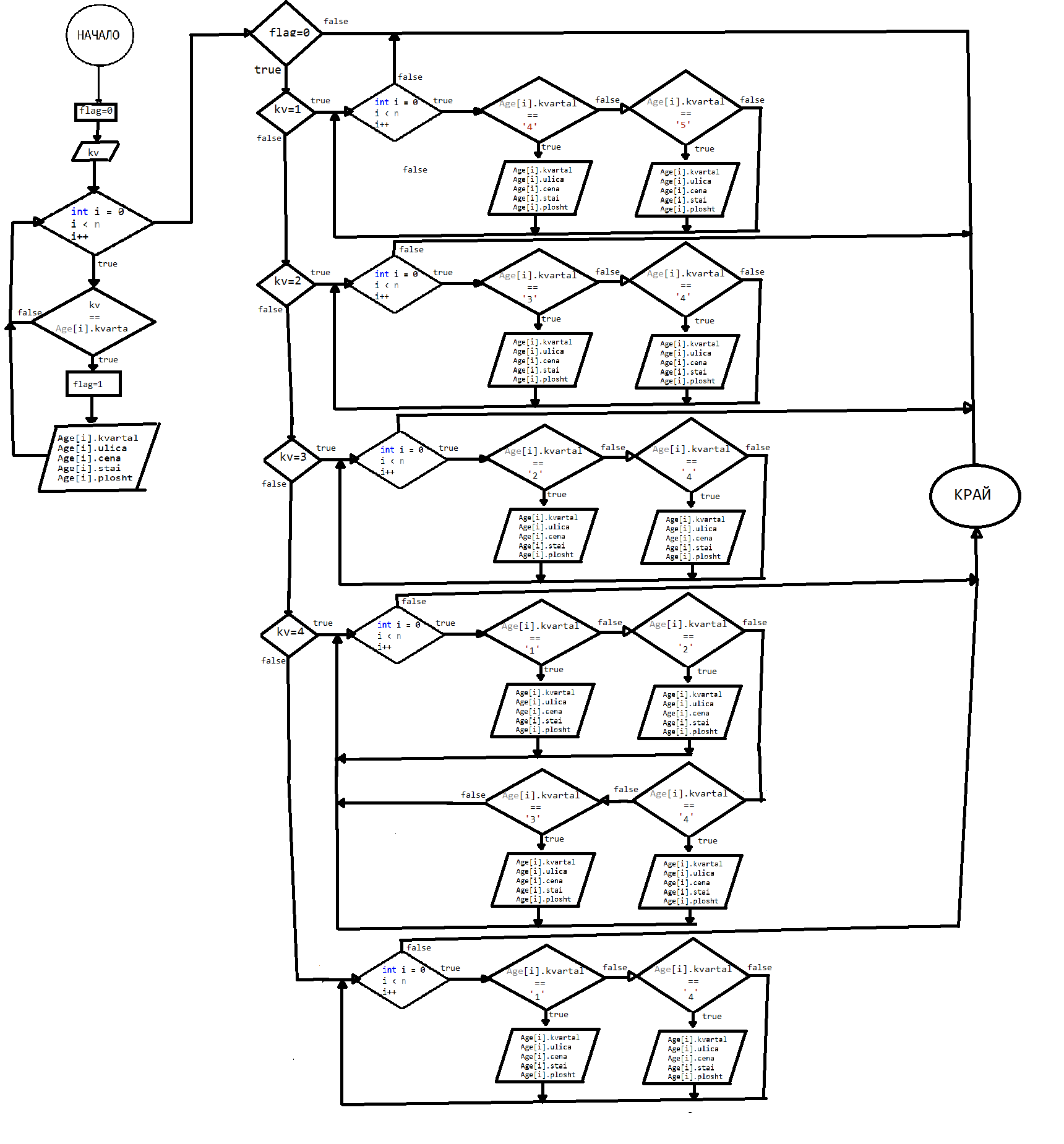
void zakupuvane(imot Age[], int n)-търси имот по улица, след което се закупува и се премахва от списъка с имоти



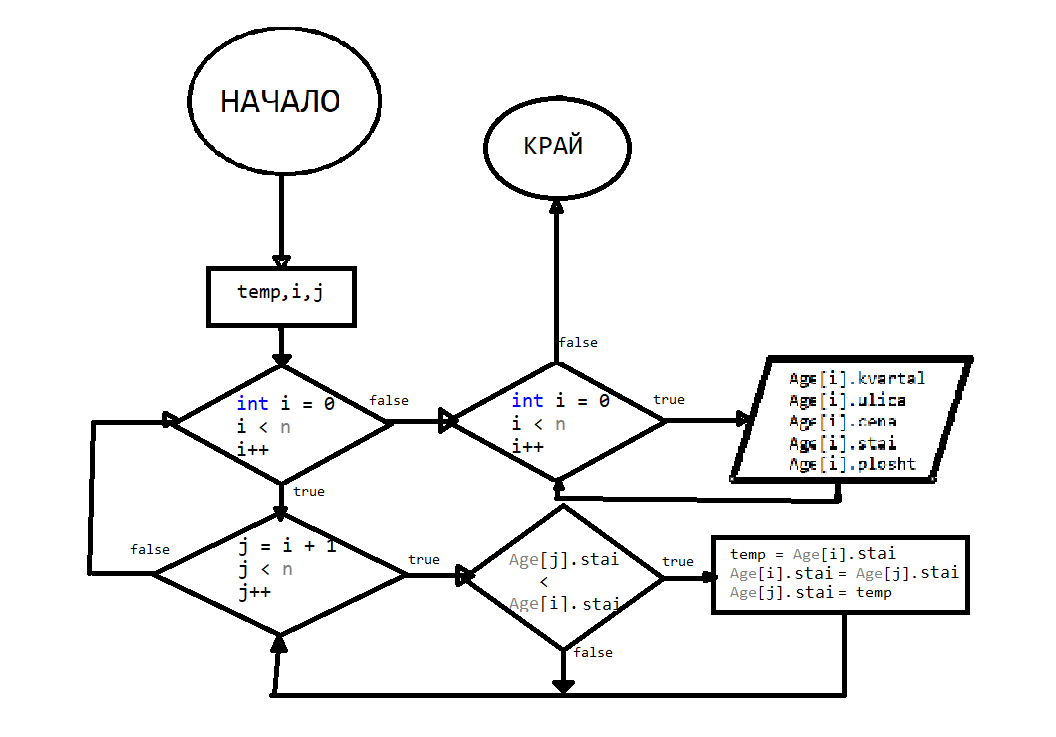
void spravka\_cena(imot Age[],int n)-извежда имотите по цена от най-висока към най-ниска



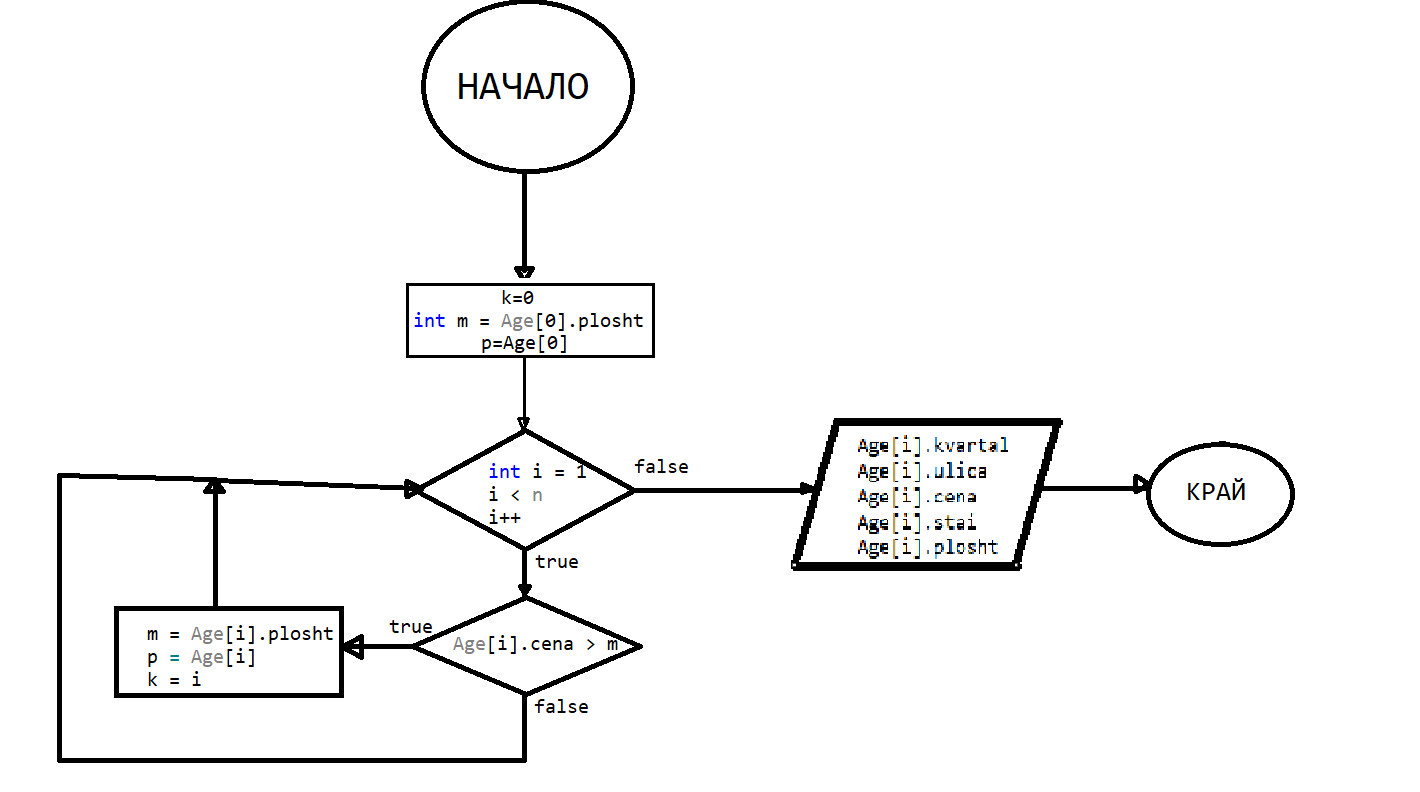
void search\_kvartal(imot Age[], int n)-търси имот по квартал, ако няма имот в квартала се извеждат имоти от съседните квартали



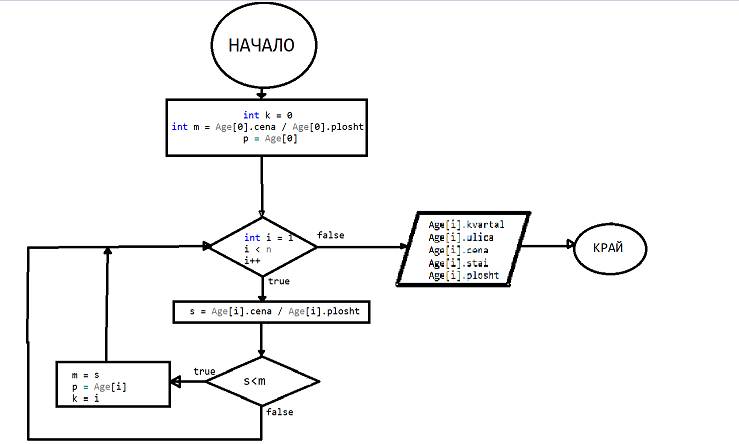
void spravka\_stai(imot Age[], int n)-извежда имотите по брой стаи от най-много стаи към най-малко



void spravka\_kvadratura(imot Age[], int n)-извежда имота с най-голяма квадратура



void spravka\_cena\_kv(imot Age[], int n)-извежда най-евтиният имот по цена на квадрат



void loadfile(imot Age[],int& n)- зарежда записан файл, ако няма такъв се извежда съобщение

void filesave(imot Age[], int n)-създава и запазва въведените данни във файл

