Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „Софтуерни и интернет технологии“

**СЕМЕСТРИАЛНА ДОМАШНА РАБОТА**

по дисциплината „Базово програмиране”

на тема: „Продажба на мобилни телефони”

Вариант 55

|  |  |
| --- | --- |
| Изготвил: Стелиан Христов | Проверил: |
| Специалност: СИТ |  |
| Група: 2Б |  |
| Факултетен номер: 22621629 |  |

2023

Съдържание

[I. Задание на проекта 3](#_Toc120821980)

[II. Анализ на решението 4](#_Toc120821981)

[1. Структура за данните в програмата 4](#_Toc120821982)

[2. Реализация на условие A 4](#_Toc120821983)

[3. Реализация на условие B 4](#_Toc120821984)

[4. Реализация на условие C 5](#_Toc120821985)

[5. Реализация на условие D 5](#_Toc120821986)

[6. Реализация на условие E 6](#_Toc120821987)

[7. Реализация на условие F 7](#_Toc120821988)

[8. Реализация на условие … - допълнение първо 7](#_Toc120821989)

[9. Реализация на условие … - допълнение второ 8](#_Toc120821990)

[10. Реализация на допълнение трето 8](#_Toc120821991)

[III. Упътване за употреба 9](#_Toc120821992)

[1. Впишете съответната част от проекта 9](#_Toc120821993)

[2. Впишете съответната част от проекта 9](#_Toc120821994)

[IV. Примерно действие на програмата 10](#_Toc120821995)

[1. Условие A 10](#_Toc120821996)

[2. Условие B 10](#_Toc120821997)

[3. Условие C 10](#_Toc120821998)

[4. Условие D 10](#_Toc120821999)

[5. Условие E 10](#_Toc120822000)

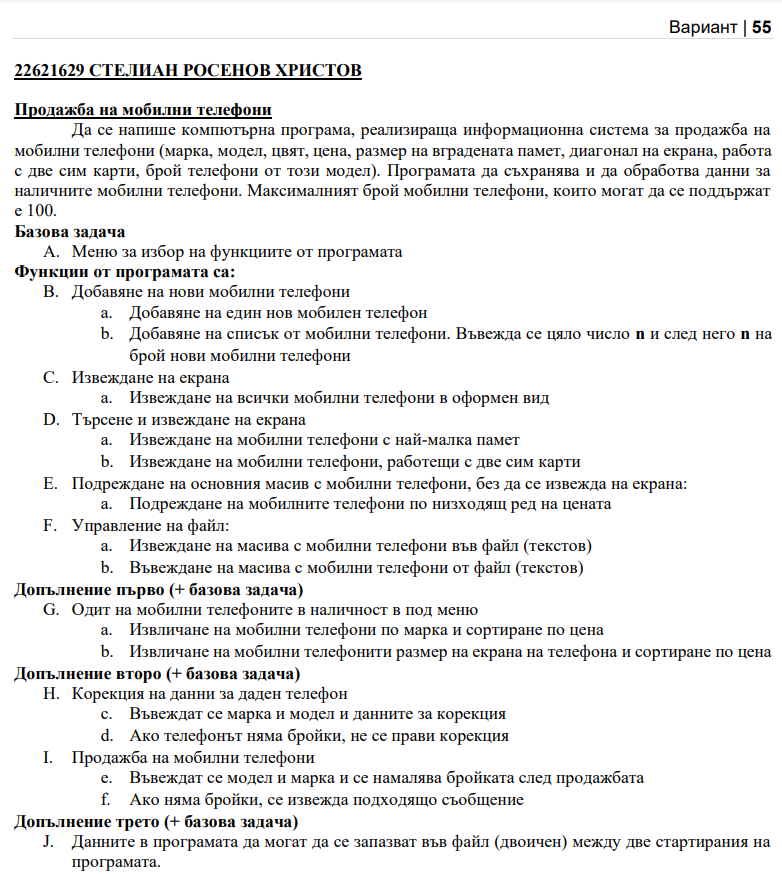
[6. Условие F 11](#_Toc120822001)

[7. Допълнение първо 11](#_Toc120822002)

[8. Допълнение второ 11](#_Toc120822003)

[9. Допълнение трето 11](#_Toc120822004)

Задание на проекта



Анализ на решението

Структура за данните в програмата

|  |  |  |
| --- | --- | --- |
| Структура | Обяснение | Примерени стойности |
| struct Phome {  string brand;  string model;  string color;  bool dualSim;  int countFromThisPhone;  double screenSize;  double price;  }; | * brand – Съхранява марката на телефона * model – Съхранява модела на телефона * color – Съхранява цвета на телефона * dual – Съхранява дали телефона поддържа две SIM карти като се пише в конзолата (yes/no) * countFromThisPhone – Съдържа броя на телефони от тази марка и модел * screenSize – Съдържа размера на екрана на телефона * price – Съдържа цена на телефона | Huawei  P30 Lite  Black  Yes  100  6.15  450 |
| Struct PhonesList {  Phone phones[100];  int size = 0;  Phone add(Phone phone){  Phones[size] = phone;  size++;  return phone;  }  }; | * phones[100] – Масив, който съхранява всички телефони. * add(Phone phone) – добавя телефон в масива и увеличава номера в масива; |  |

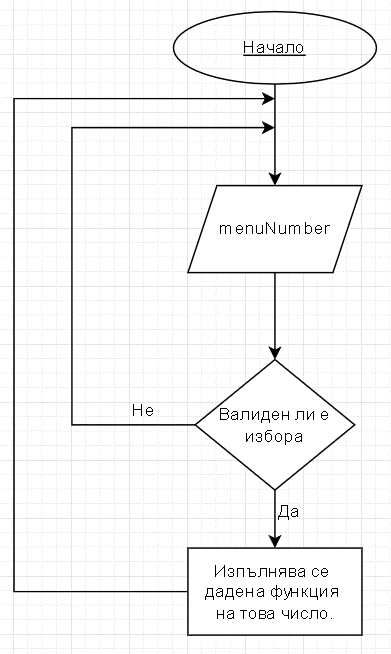
Реализация на условие A

Анализ на алгоритъма, който трябва да се реализира

Създаваме просто меню с 9 опции.

Потребителя избира функция като изписва в конзолата число от 1 до 9. Ако изпише нещо друго различно. Му се извежда пак менюто за да избере отново.

Блок схема на алгоритъма



Функция, с която е реализиран алгоритъма

void menu(Phone phones[100], PhonesList list);

#### Входни данни на функцията

Декларира се цяло число, в което се запаметява избора на потребителя. И един символ, който се променя при опит за изход от програмата

#### Изходни данни на функцията или данни, които се извеждат

Функцията не извежда данни.

Реализация на условие B

Анализ на алгоритъма, който трябва да се реализира

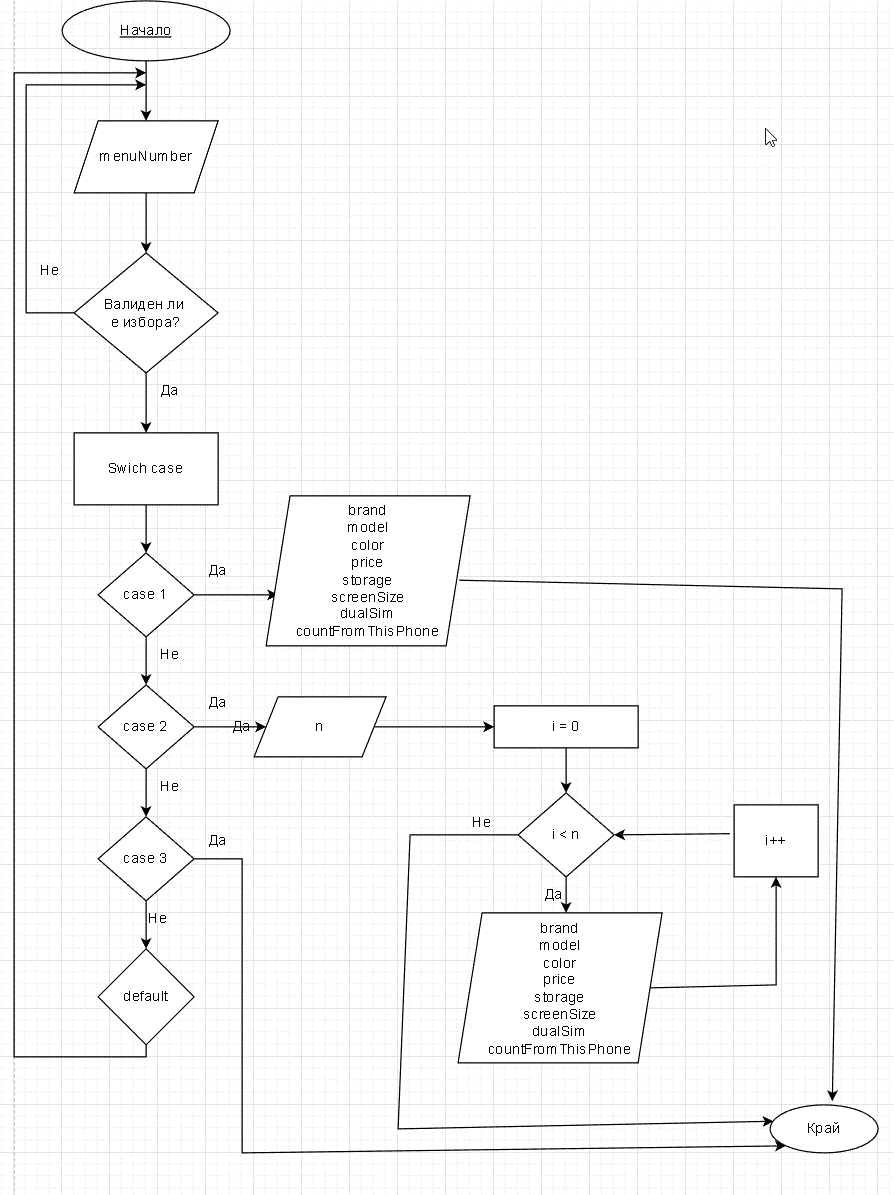
Създаваме просто подменю с 3 опции:

Да се добави един мобилен телефон.

Да се добави списък с телефони.

Потребителя избира дали иска да добави един телефон или списък телефони. Също така има избор за изход от подменюто.

Блок схема на алгоритъма



Функция, с която е реализиран алгоритъма

void addPhoneMenu(Phone phones[100], PhonesList& list);

void phoneFromInput(PhonesList& list);

vod phonesFromInput(PhonesList& list);

#### Входни данни на функцията

Потребителя ако избере да добави само един телефон трябва да въведе марка, модел, цвят, цена, размер на паметта, размер на екрана (в инчове), работа с две СИМ карти и бройки от този телефон. Ако избере множество телефони, трябва да въведе колко телефона желае да добави. След това трябва да запише на всеки телефон марка, модел, цвят, цена, размер на паметта, размер на екрана (в инчове), работа с две СИМ карти и бройки от този телефон.

#### Изходни данни на функцията или данни, които се извеждат

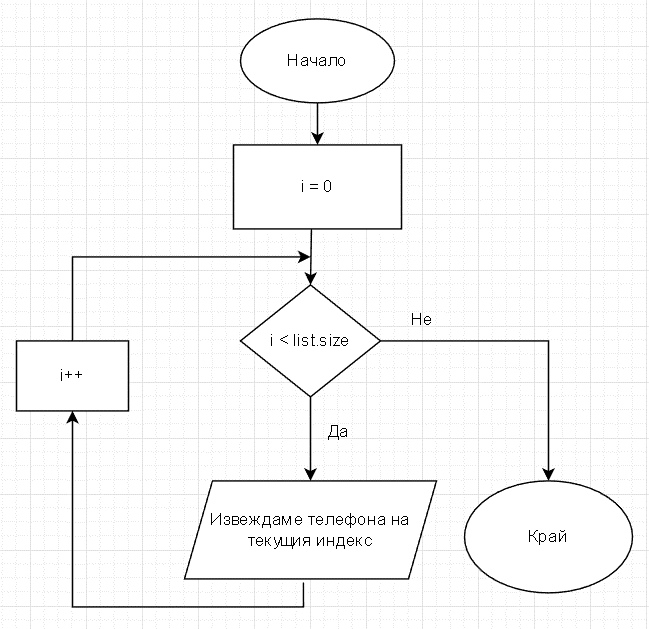
Изписва, че дадения телефон е добавен.

Реализация на условие C

Анализ на алгоритъма, който трябва да се реализира

След като потребителя е избрал тази функция се обхожда масива и се извеждат наличните телефони в интерфейс приличащ на таблица, а ако няма налични телефони се отпечатва подходящо съобщение.

Блок схема на алгоритъма



Функция, с която е реализиран алгоритъма

void showPhones(PhonesList& list);

#### Входни данни на функцията

Функцията не приема входни данни.

#### Изходни данни на функцията или данни, които се извеждат

При налични телефони се извежда таблица с наличните телефони, в противен случай се извежда съобщение „Базата данни е празна!“.

Реализация на условие D

Създаваме просто подменю с 3 опции:

Да се намери телефони с най-малко памет.

Да се намерят телефони, които поддържат две СИМ карти.

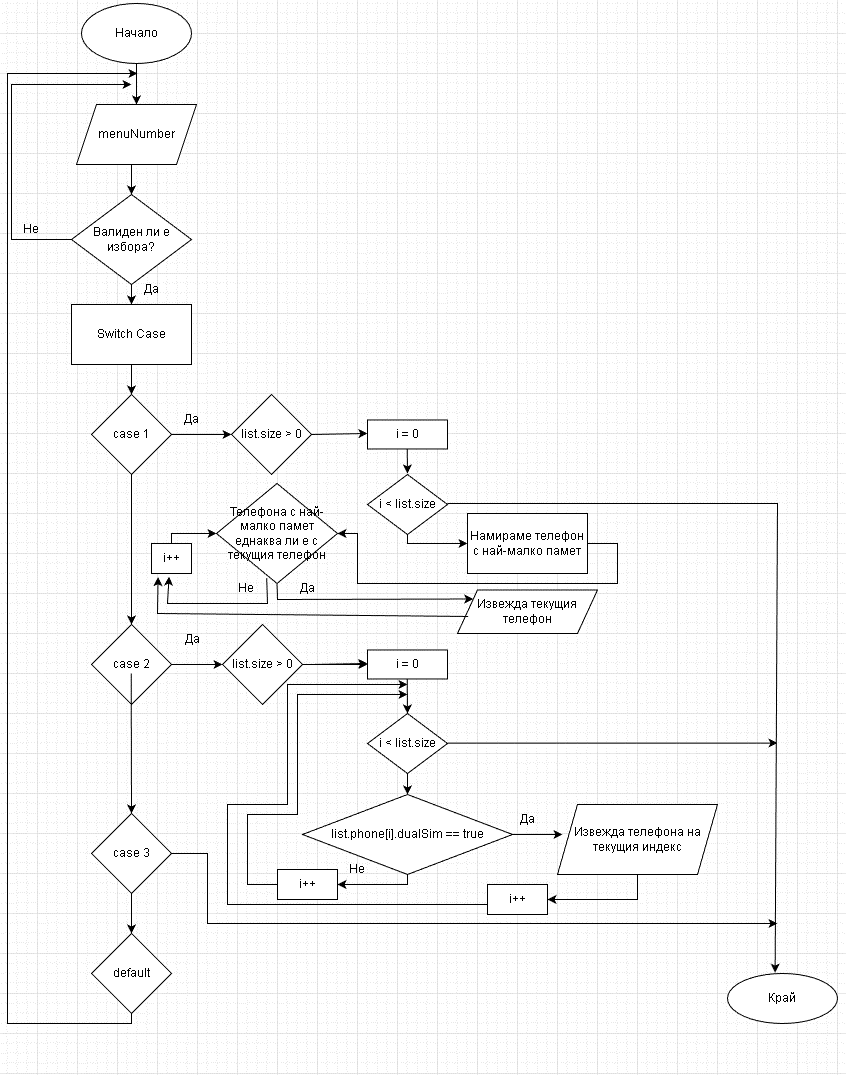
Изход

Анализ на алгоритъма, който трябва да се реализира

За да намерим телефоните с най-малко памет обхождаме масива и намираме първо телефон с най-малко памет, след това обхождаме масива и търсим телефон със същата памет и отпечатваме.

За да намерим телефони, които поддържат две СИМ карти обхождаме масива и проверяваме дали текущия телефон променливата му dualSim ако е true отпечатваме.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void searchAndPrintMenu(Phone phones[100], PhonesList list);

void smallStorage(PhonesList list);

void supportDualSIM(PhonesList list);

#### Входни данни на функцията

phones и list. phones е масив от 100 елемента. А list e структура, с която можем да достъпим размер на масива(list.size) и можем да достъпим масива с телефони(list.phones[i]). Използваме ги за да открием телефона с най-малка памет и телефони поддържащи две СИМ карти.

#### Изходни данни на функцията или данни, които се извеждат

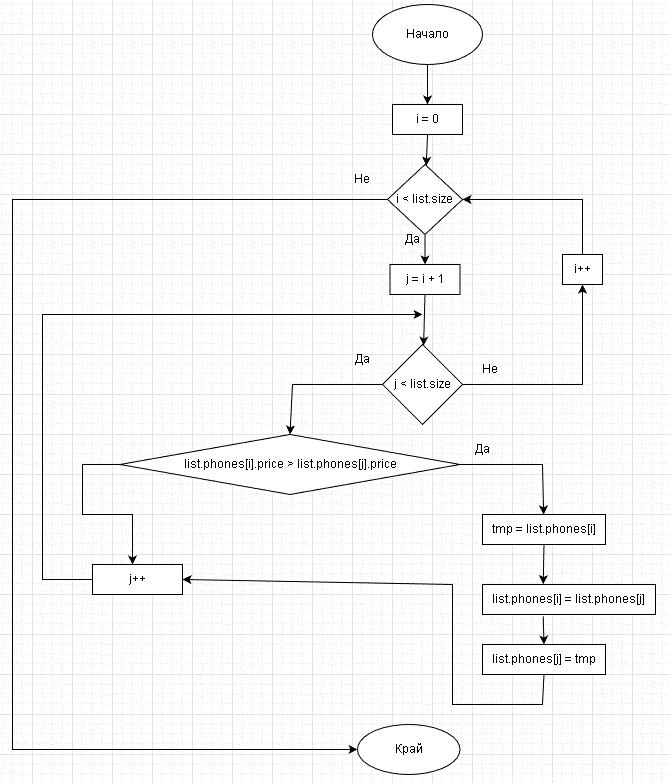
Изходните данни са на вид на таблица или вид на съобщение. В случай, че има телефони извежда телефона с най-малка памет във вид на таблица, в случай че няма се извежда съобщение, че няма налични телефони. Ако има телефони, които поддържат две СИМ карти се извеждат във вид на таблица, в противен случай се извежда съобщение, че няма телефони които поддържат две сим карти.

Реализация на условие E

Анализ на алгоритъма, който трябва да се реализира

Подреждаме масива с телефони по цена низходящ ред, без да го отпечатваме. Като сравняваме всеки два елемента, ако левия е по-голям ги разменяме. Така големите елементи изплуват надясно като балончета. Като сравняваме всеки два елемента и ако левия елемент е по-голям ги разменяме. Така големите елементи изплуват вдясно като балончета. Защото не е нужно да връщаме списъка с телефоните. Данните НЕ се извеждат.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void sorting(PhonesList& list);

#### Входни данни на функцията

Функцията не приема входни данни.

#### Изходни данни на функцията или данни, които се извеждат

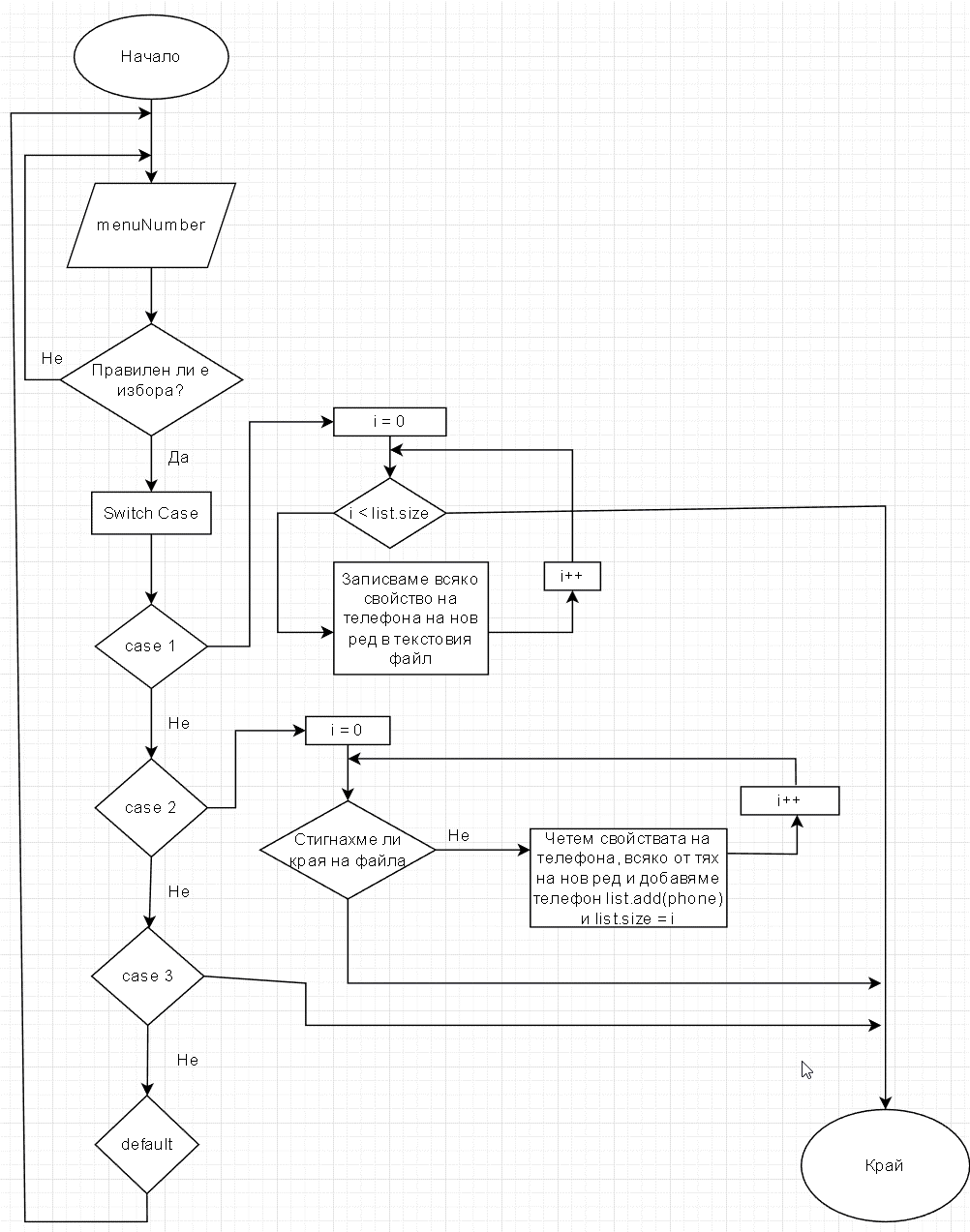
Функцията не извежда данни.

Реализация на условие F

Анализ на алгоритъма, който трябва да се реализира

Извежда се подменю, от което потребителя избира дали да запази данните или да зареди данните от файл има и изход от подменюто. Запазването създава файл с име “telefoni.txt”. И започва обхождане на масива с телефони като всяка данна е на нов ред. Зареждането на данни става като търси файл с име “telefoni.txt”, ако не го намери извежда подходящо съобщение. След като бъде отворен започва да чете всеки ред докато не стигне края на файла като запазва всяка данна на мястото си както е направен записа така и четенето. Като при всяко прочитане на телефон се добавя телефона в масива с помощта на list.add(phone);

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void controllFilesMenu(Phone phones[100], PhonesList& list);

void printOnTextFile(PhonesList& list);

void readFromTextFile(PhonesList& list);

#### Входни данни на функцията

Няма входни данни. Освен избора на функция.

#### Изходни данни на функцията или данни, които се извеждат

Изходните данни се запазват в текстов файл.

Реализация на условие … - допълнение първо

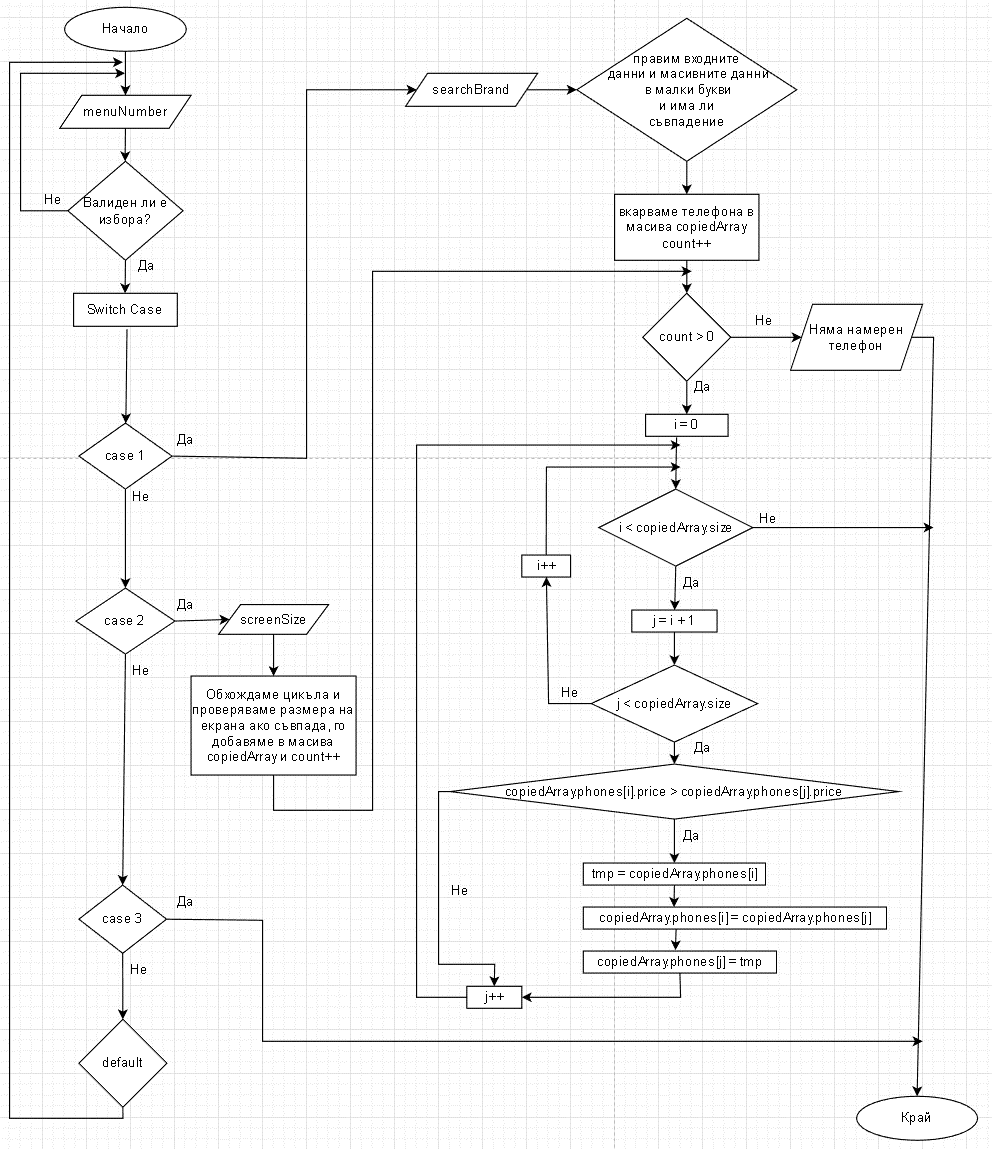
Анализ на алгоритъма, който трябва да се реализира

Извежда се подменю за избор дали иска да провери телефони по марка или по размер на екрана и сортиране по цена. Също така има и изход.

За проверка на телефони по марка. Декларираме променлива tmp от тип Phone също така listOfPhones(който е копие на оригиналния масив с телефони), copiedArray(празен масив) от тип PhonesList и searchBrand от тип string.  
Потребителя въвежда марката без значение дали буквите са главни или малки. Входните данни и списъка с телефони на масив правим бувките малки и проверяваме дали има съвпадение, ако има съвпадение от оригиналния масив вкарваме в масива копие телефона също ако количеството е повече от 0. Ако не бъде намерен такъв телефон се извежда подходящо съобщение.

За проверка на телефони по размер на екрана обхождаме целия масив и търсим телефон с такъв какъвто е написал потребителя. Ако не бъде намеренето се извежда подходящо съобщение. И след това се подреждат по възходящ ред по цена.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void fetchPhonesMenu(PhonesList& list);

void fetchPhonesByBrand(PhonesList& list);

void fetchPhonesByScreenSize(PhonesList& list);

#### Входни данни на функцията

Избор на опция – цяло число от 1 до 3 тип int

Извличане на телефон по марка – въвежда се марка на телефона(текст от тип string)

Извличане на телефон по размер на екрана – въвежда се размер на екрана в инчове(реално число от тип double)

#### Изходни данни на функцията или данни, които се извеждат

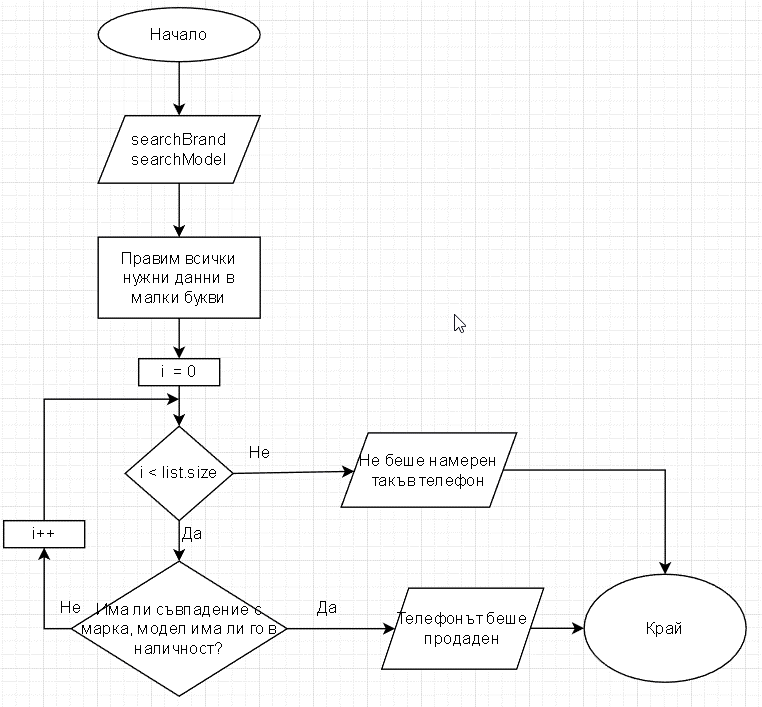
Ако намери съвпадение, данните се извеждат във вид на таблица в противен случай се извежда подходящо съобщение.

Реализация на условие … - допълнение второ

Анализ на алгоритъма, който трябва да се реализира

Продажба на телефон. Потребителя въвежда марка и модел на телефона. Данните на потребителя се преобразуват в малки бувки също и в масива копие ако има сравнение се изписва съобщение, че се е продал даден телефон и бройката му се намаля. В противен случай, когато няма в наличност се извежда подходящо съобщение.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void sellPhone(PhonesList& list);

#### Входни данни на функцията

Входните данни са тип string.

string searchBrand;

string searchModel;

#### Изходни данни на функцията или данни, които се извеждат

При успешна продажба се изписва съобщение, че телефона е продаден. При изчерпано количество на даден телефон се изписва подходящо съобщение.

Реализация на допълнение трето

Анализ на алгоритъма, който трябва да се реализира

Данните от масива се преобразуват в двоичен формат и се записват във файл с име “database.dat“. Ако възникне грешка се изписва, че е възникнала грешка.

Когато бъде стартирана програмата, тя търси файл с име “ database.dat“ и го отваря и чете телефоните от там докато стигне края на файла. В противен случай се извежда съобщение грешка.

Блок схема на алгоритъма

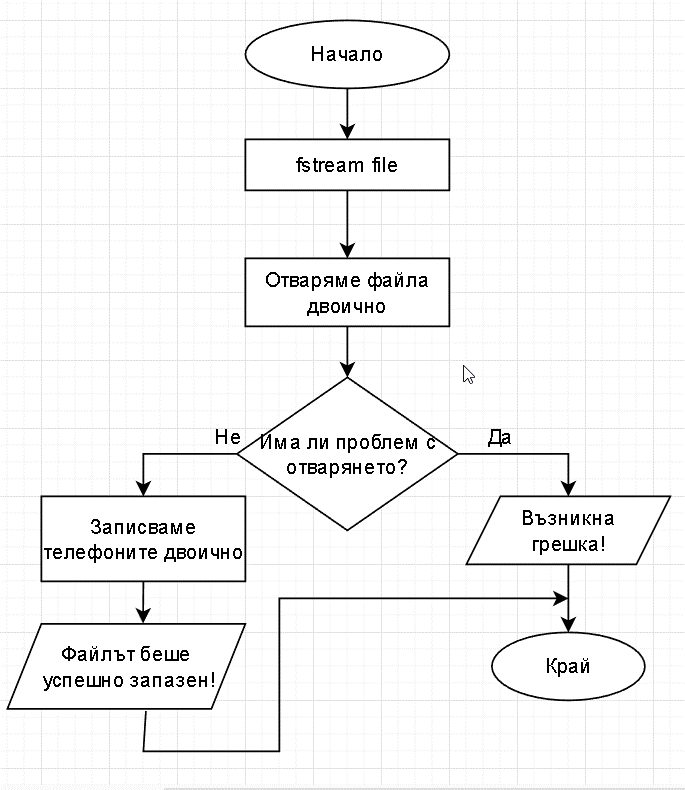
Тук се поставя блок схема на описания алгоритъм.

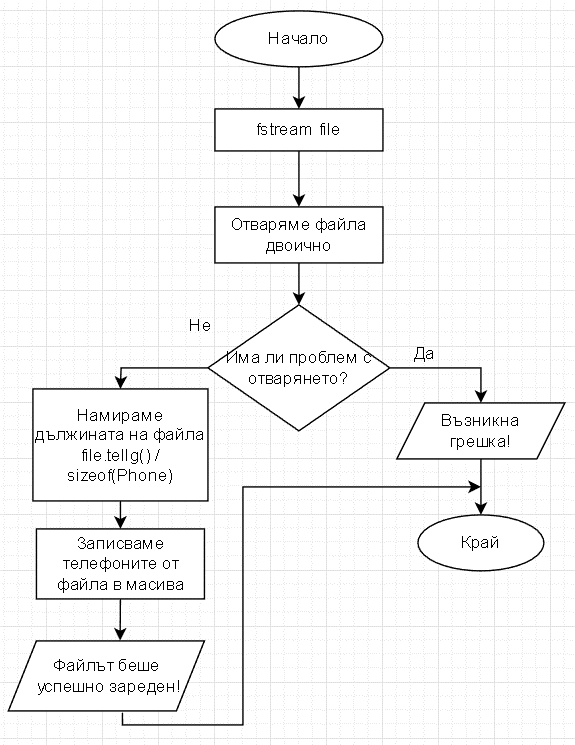
Функция с която е реализиран алгоритъма

void saveBinaryFile(PhonesList& list);

void loadBinaryFile(PhonesList& list);

#### Входни данни на функцията





#### Изходни данни на функцията или данни, които се извеждат

При успешно запазване на файла се изписва, че файлът е запазен в противен случай, се извежда подходящо съобщение.

Упътване за употреба

Представете какво трябва да се въведе за правилното използване на програмата, какво се извежда в резултат на нея. Какви са очакваните резултати и какво ще се случи, ако се въведат невалидни данни. Изтрийте този текст, след като се запознаете с него.

## **Условие А**

Очаквани входни данни

Цяло число от 1 до 9

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Да се изпълни избраната функция.

## **Условие B**

Очаквани входни данни

Цяло число от 1 до 3 – Избор дали искате да добави един телефон, списък от телефони или да напуснете функцията.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Когато бъде добавен телефона се изписва, че е добавен телефон с марка и модел.

## **Условие C**

Очаквани входни данни

Не изисква входни данни.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

При наличие на мобилни телефони ще ги изведе в оформен вид като таблица, в противен случай ще изведе подходящо съобщение.

## **Условие D**

Очаквани входни данни

Цяло число от 1 до 3 – Избор за намиране на телефони с най-малко памет, телефони които поддържат две СИМ карти или изход от функцията.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

При наличие на мобилни телефони ще ги изведе в оформен вид като таблица, в противен случай ще изведе подходящо съобщение.

## **Условие E**

Очаквани входни данни

Не изисква входни данни.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Не се извеждат изходни данни, но се подреждат телефоните.

## **Условие F**

Очаквани входни данни

Цяло число от 1 до 3 – Избор дали телефоните да се запазят в текстов файл, заредят от текстов файл или изход от функцията.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

При успешна операция се изписва подходящо съобщение, а при неуспешна операция се изписва “Възникна грешка!“.

## **Условие G**

Очаквани входни данни

Цяло число от 1 до 3 – Избор дали да търсите по марка, дали да търсите по размер на екрана или изход от функцията.

При търсене по марка се изисква изписване на марката на телефона.

При търсене по размер на екрана се изисква изписване размера на екрана на телефона.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Ако не бъде намерен телефон ще бъде изписано, че не е намерен телефон. А при успеншо намиране ще се изведе във вид на таблица и подредено по цена.

## **Условие H**

Очаквани входни данни

Въвежда се марка и модел на телефона.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Ако не бъде открит телефона се изписва съобщение, че не съществува такъв телефон в базата данни. Ако от този телефон няма бройки се изписва, че няма бройки и няма право да корекция.

## **Условие I**

Очаквани входни данни

Въвежда се марка и модел на телефона.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Ако не бъде открит телефона се изписва съобщение, че не съществува такъв телефон в базата данни. Ако от този телефон няма бройки се изписва, че няма бройки и няма право да корекция. А ако има бройки се изписва съобщение, че е продаден този телефон и бройките му се намаляват с едно.

## Условие J

Очаквани входни данни

Не се изискват входни данни.

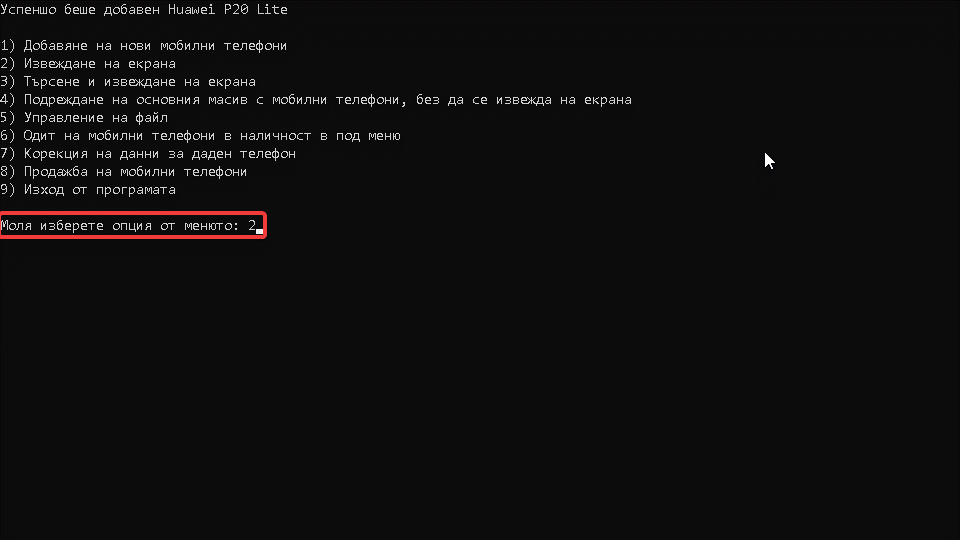
Очакван резултат от изпълнението на конкретния фрагмент от проекта

При успеншо запазване/зареждане се извежда подходящо съобщение, в противен случай се извежда, че е възникнала грешка.

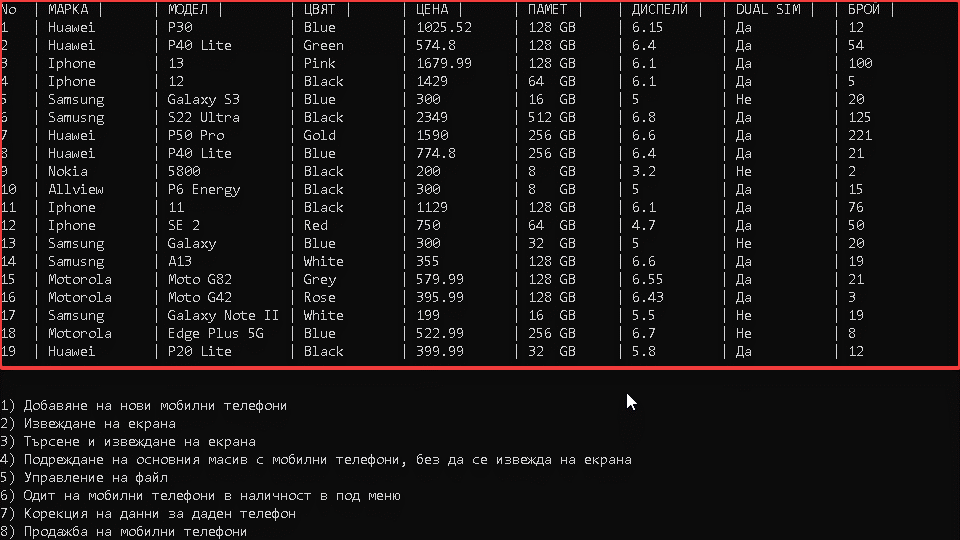
Примерно действие на програмата

Условие A

Снимка на изгледа с примерни входни данни

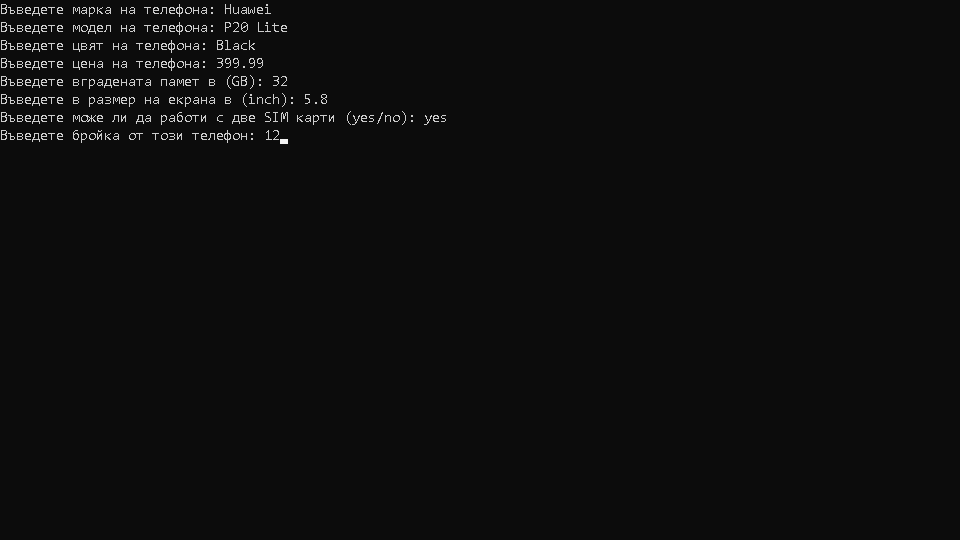


Снимка на изгледа с примерни изходни данни

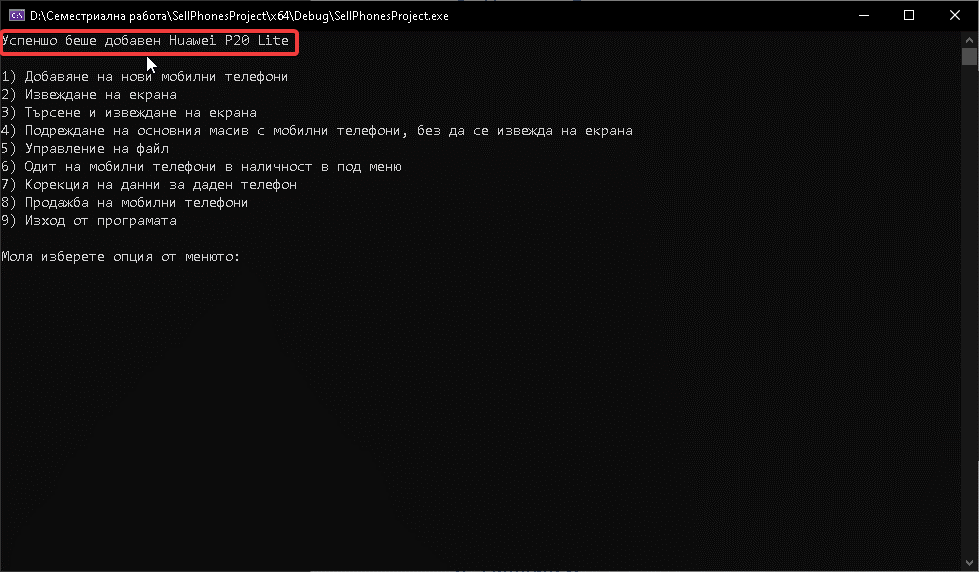


Условие B

Снимка на изгледа с примерни входни данни

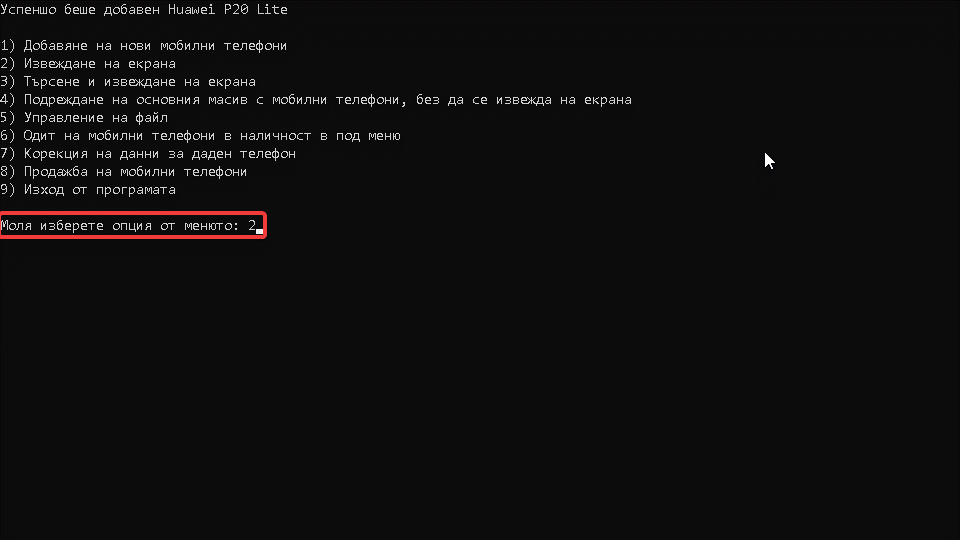


Снимка на изгледа с примерни изходни данни

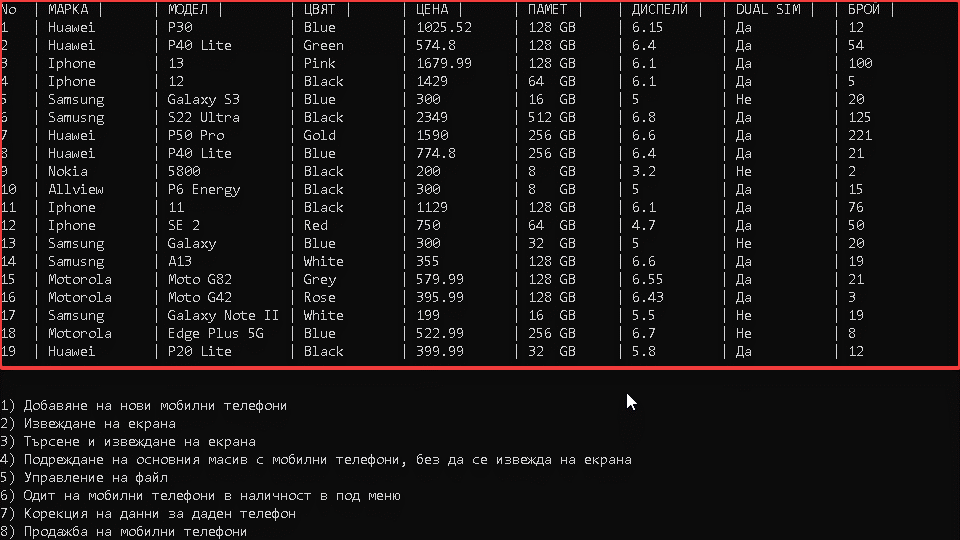


Условие C

Снимка на изгледа с примерни входни данни

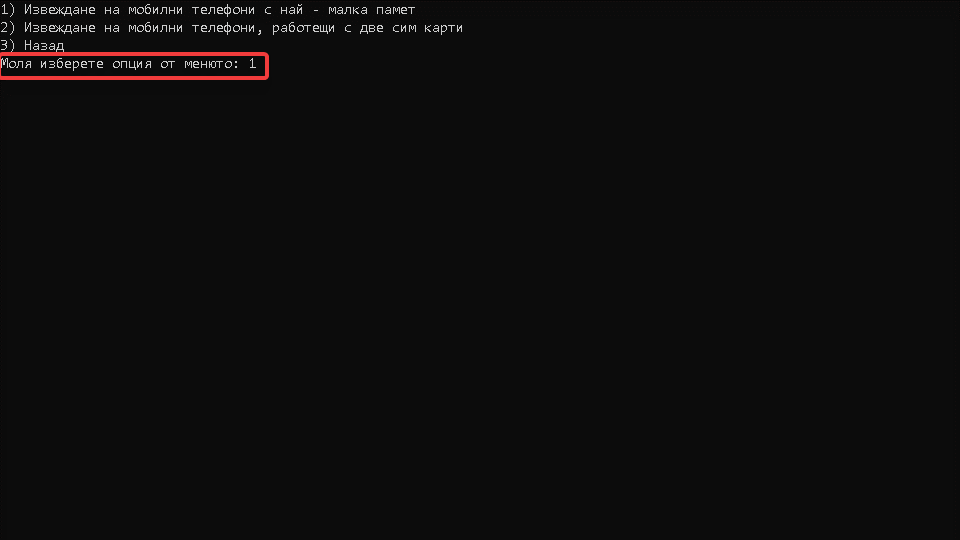


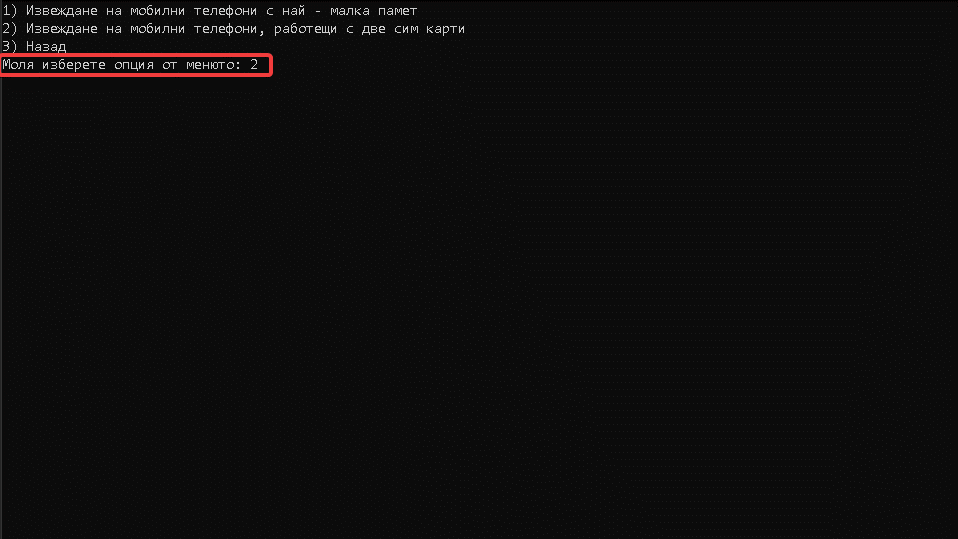
Снимка на изгледа с примерни изходни данни



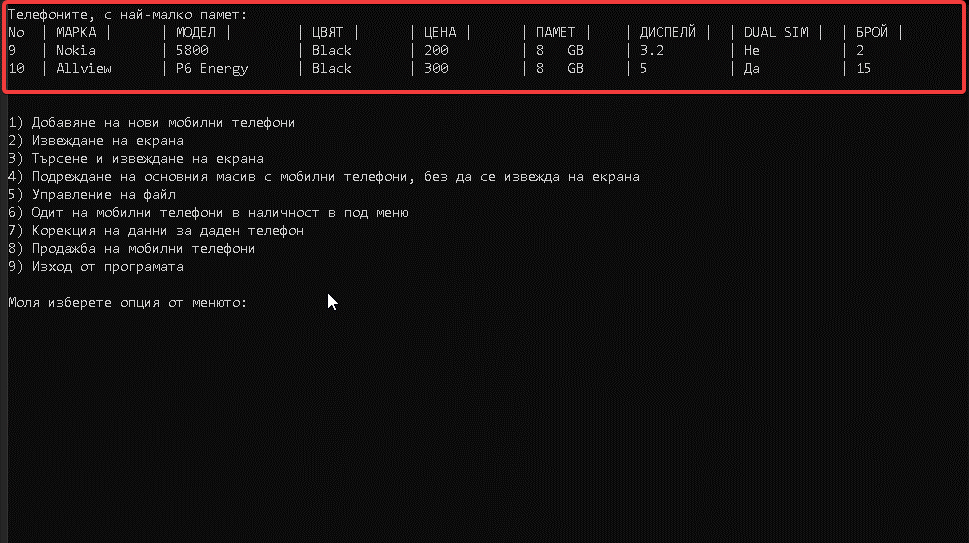
Условие D

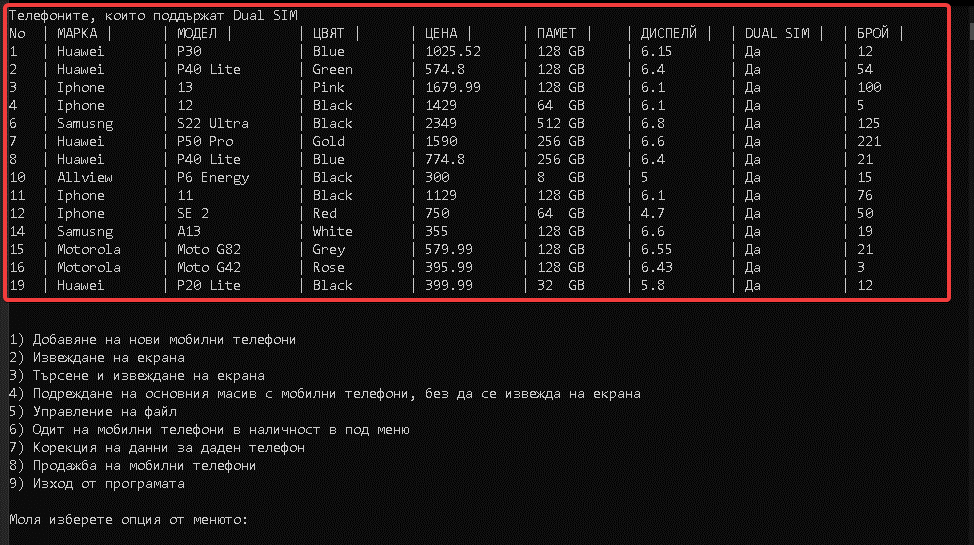
Снимка на изгледа с примерни входни данни





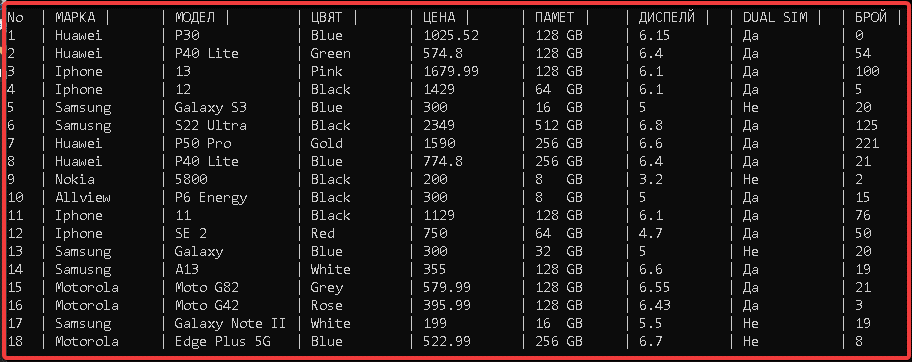
Снимка на изгледа с примерни изходни данни



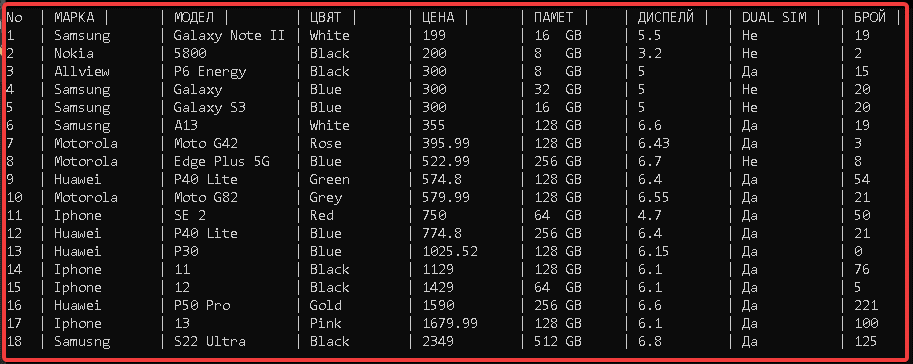


Условие E

Снимка на изгледа с примерни входни данни

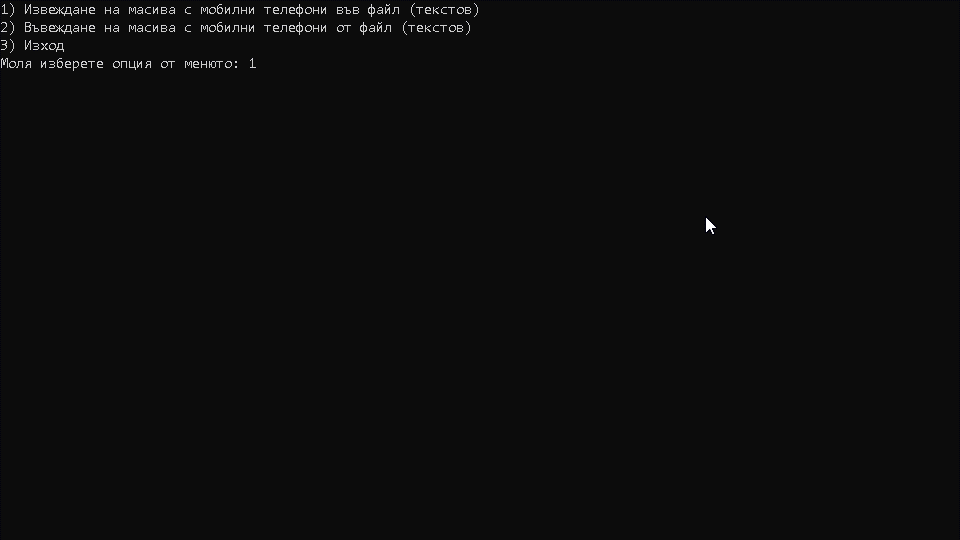


Снимка на изгледа с примерни изходни данни

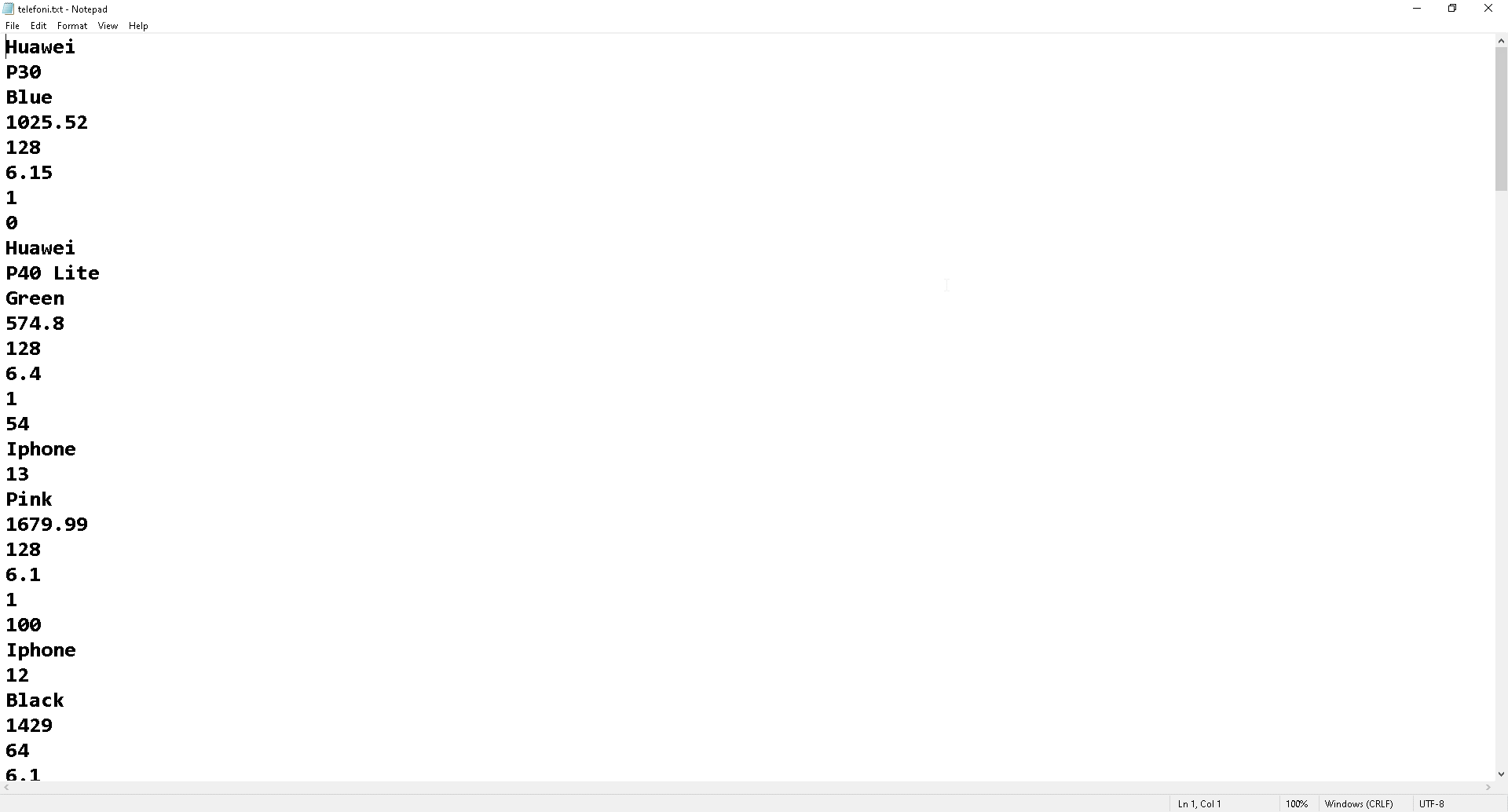


Условие F

Снимка на изгледа с примерни входни данни

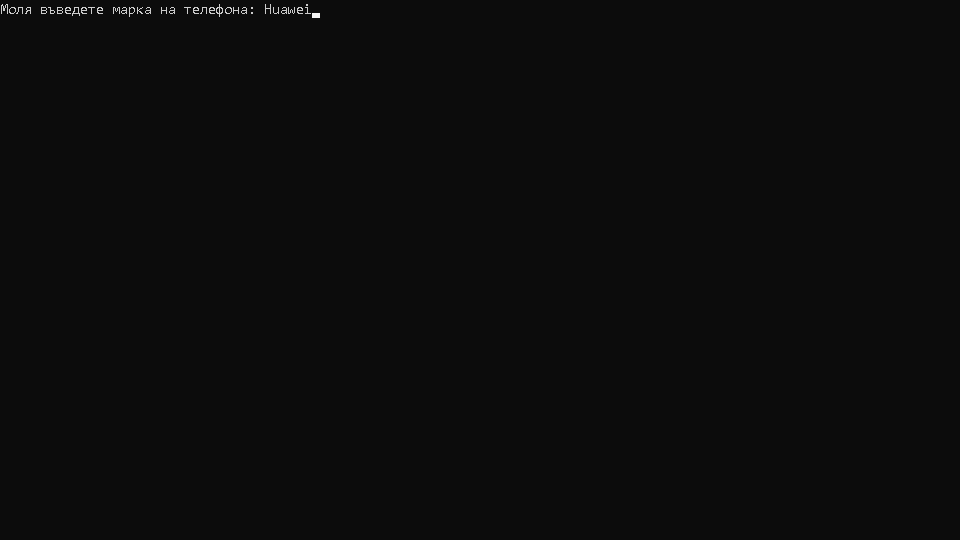


Снимка на изгледа с примерни изходни данни

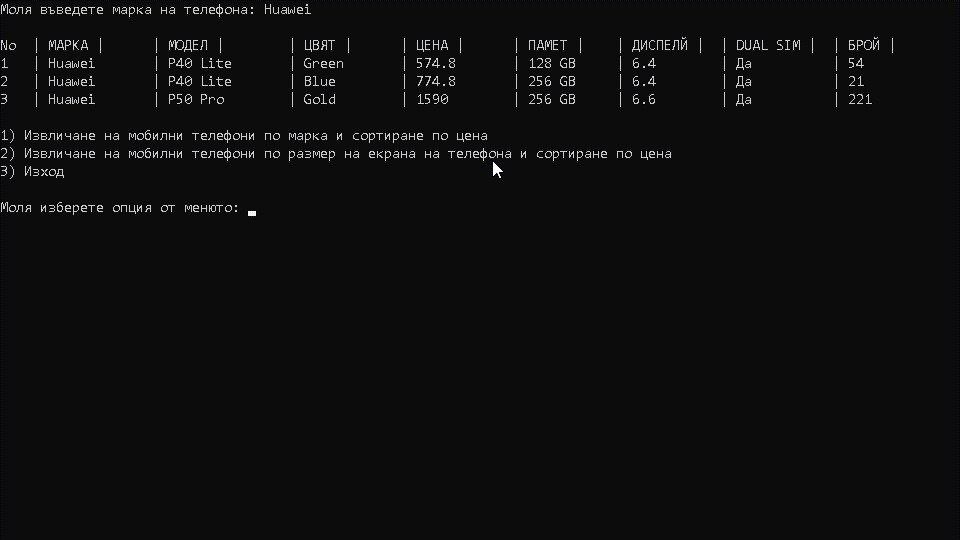


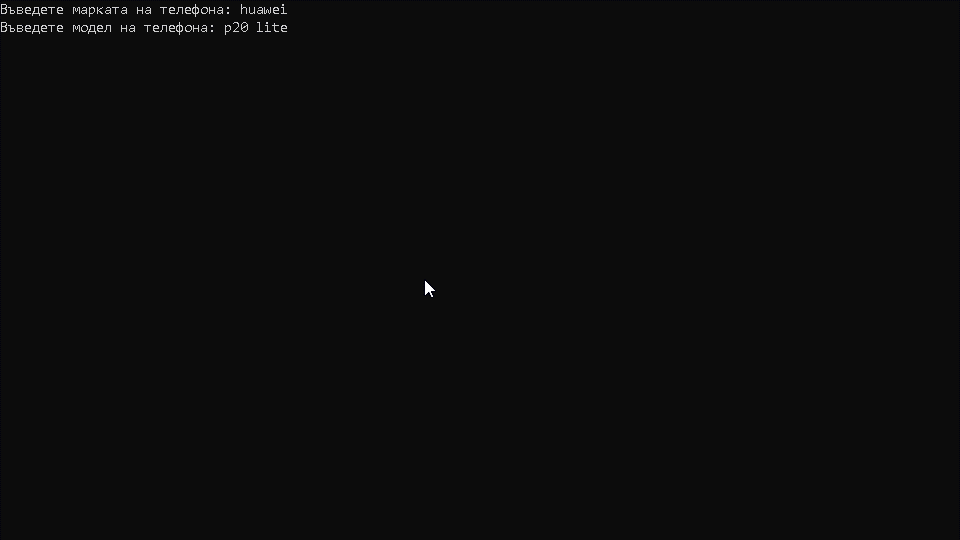
Допълнение първо

Снимка на изгледа с примерни входни данни



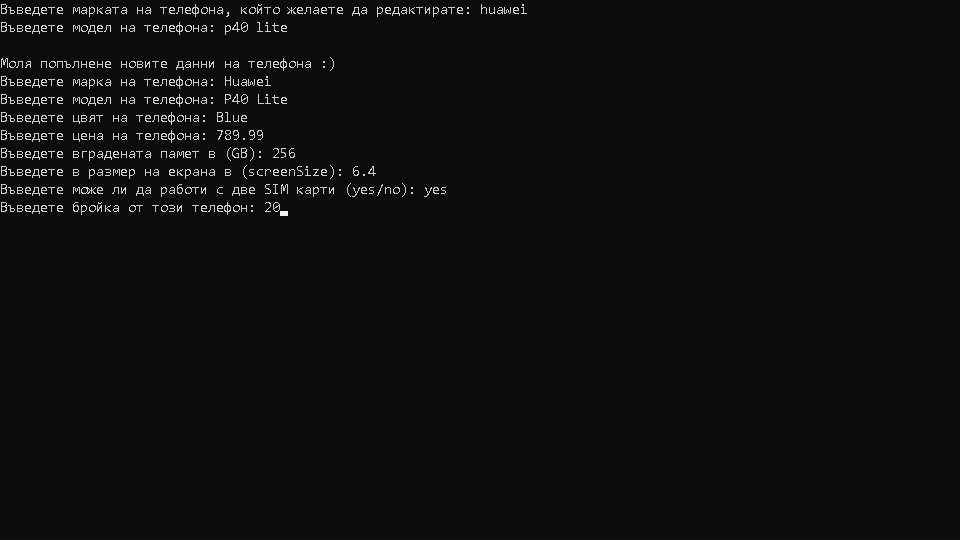
Снимка на изгледа с примерни изходни данни

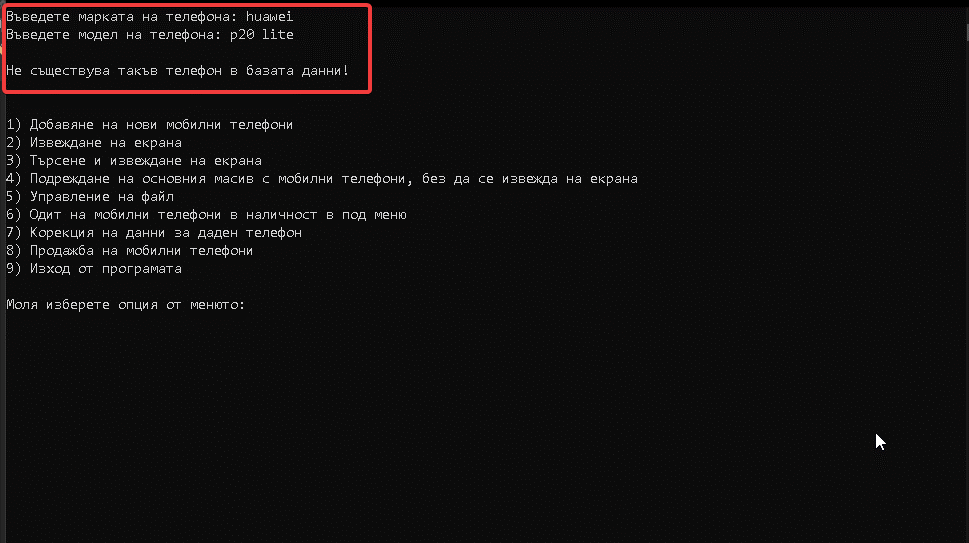




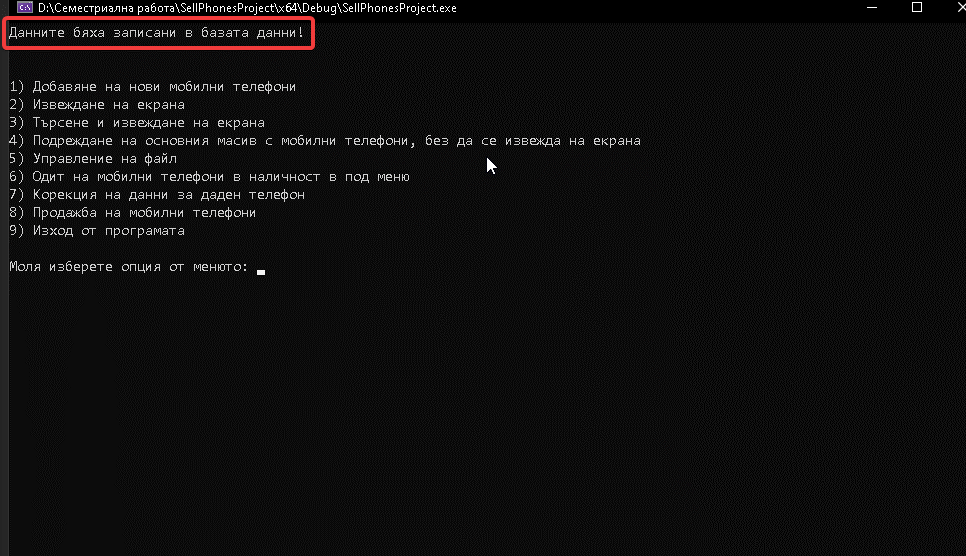
Допълнение второ

Снимка на изгледа с примерни входни данни



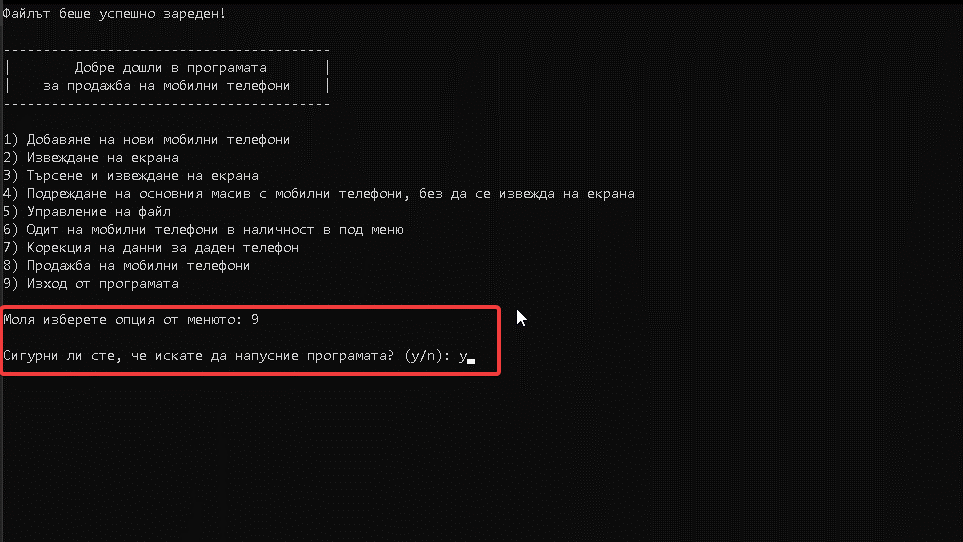


Снимка на изгледа с примерни изходни данни



Допълнение трето

Снимка на изгледа с примерни входни данни



Снимка на изгледа с примерни изходни данни

