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: „Информационна система Студенти”

Вариант 241

|  |  |
| --- | --- |
| Изготвил: Гилгамеш Алджабари | Проверил: |
| Специалност: СИТ |  |
| Група: 3 |  |
| Факултетен номер: 23621654 |  |

2024

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 G - допълнение първо 7](#_Toc120821989)

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

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

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

[1. Упътвания за употреба за условия от A до F 9](#_Toc120821993)

[2.Упътвания за употреба за допълнения от G до J 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

**Информационна система „Студенти“**

Да се напише компютърна програма, реализираща информационна система, която поддържа студентска група (факултетен номер, ЕГН, три имена, списък с дисциплини, пол, възраст, статут-активен/прекъснал/завършил). Програмата съхранява и обработва данни за студентите в групата. Максималния брой студенти в една група е 30. Всеки студент има 5 оценки по дисциплините (име, оценка) от 1 семестър

**Базова задача**

1. Меню за избор на функциите от програмата.

**Функции от програмата са:**

1. Добавяне на студенти в група:

а. Добавяне на студенти в групата, чрез добавяне в масива, като не трябва да се превишава максималният брой студенти в една група (30).

1. Извеждане на всички студенти на екрана

а.Извеждане на всички студенти, чрез подходящо форматиране  в таблица

D. Търсене и извеждане на екрана:

а. Търсене и извеждане на студенти с оценка по БП в определен в интервал [min – max]

b. Търсене и извеждане на студенти с минимум един предмет, с оценка Слаб 2

E. Сортиране и извеждане на студенти на екран:

a. Сортиране на студентите по малко име по азбучен ред

F. Работа с външен файл (двоичен или текстов)

a. Извеждане на масива със студенти във файл

b. Въвеждане на масива със студенти от файл

**Допълнение първо (+ базова задача)**

G. Да се създаде подменю към основното с нови функции за:

а. Отделяне на студенти с определен статут (въвежда се от потребителят) и сортиране по факултетен номер.

b. Отделяне на студенти с успех в интервал (мин -макс) и сортиране по ЕГН.

**Допълнение второ (+ базова задача)**

H. Въвеждане/Актуализация на оценки и среден успех за студент

а. Добавяне в структурата на поле среден успех.

b. Въвеждане/Актуализацията на данни (оценки и среден успех), става по въведен факултетен номер на студент.

i. Ако студентът присъства в списъка(масива):

1. Проверява се статута му:

a. Активен – при въвеждане/актуализация на оценки, се изчислява автоматично средният успех. Всяка промяна на оценка изисква запитване за промяна! Ако няма оценка, тя се актуализира без запитване!

b. Прекъснал/завършил – НЕ могат да се актуализират оценки

c. Ако студента не присъства в списъка(масива), се извежда пояснително съобщение на екрана.

I. Смяна на статут на студент

a. Въвежда се факултетният номер на студента(ако го има, се въвежда и новият статут). Ако статутът е активен или прекъснал се сменя с новият. Ако статутът е завършил НЕ СЕ допуска промяна!

**Допълнение трето (+ базова задача)**

**J.** Данните в програмата да се попълват автоматично от файл при стартиране и да се записват автоматично във файл при затваряне на програмата.

Анализ на решението

Структура за данните в програмата

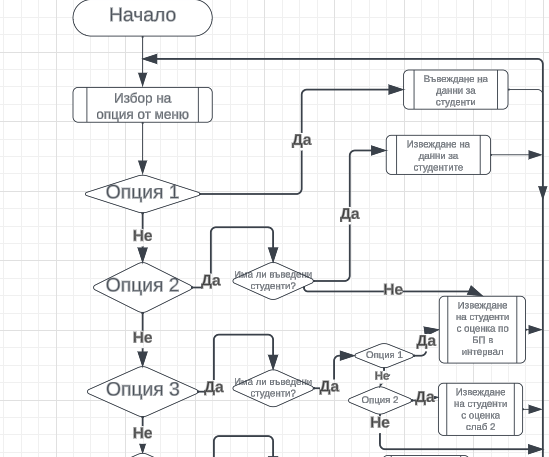
|  |  |  |
| --- | --- | --- |
| Структура | Обяснение | Примерени стойности |
| struct student\_info {  bool active;  string faculty\_number;  string personal\_id\_number;  string full\_name;  string gender;  int age;  string status\_of\_student;  double sreden\_uspeh;  discipline disciplines[5] = {  {"Basis of programming", 0},  {"Maths", 0},  {"Web Design", 0},  {"Electronics", 0},  {"Sport", 0}  };  }; | * аctive – Използва се от функцията return\_num\_students като ако се въведат данни за поредния студент променливата става true. * faculty\_number – съхранява факултетния номер * personal\_id\_number – съхранява ЕГН * full\_name – съхранява пълно име * gender – съхранява пол * status\_of\_student – съхранява статут на студента(graduated, dropout, active) * sreden\_uspeh – съхранява сметнат среден успех за студент | true  23621654  6101057509  Gilgamesh Aljabari  Male  18  active  6.00 |
| discipline disciplines[5] = {  {"Basis of programming“, 0}  {"Maths", 0},  {"Web Design", 0},  {"Electronics", 0},  {"Sport", 0}  };  struct discipline  {  string name;  int grade;  }; | * discipline[5] – масив от тип discipline с 5 члена като има един за всеки предме * Всеки предмет си има име( предефенирано) и оценка * name - съхранява името на дисциплината * grade съхранява оценката за дисциплината   Направено е за по-добра асоциация между предмет и оценка  Тази структура е помощна   * name съхранява името на предмета * grade съхранява оценката   Използва се в главната структура като се инициализира масив от 5 структури тип discipline. Предметите за всеки член на масива са написани предварително, а оценките са инициализирани със 0 | Basis of programming  6  Maths  6  Web Design  6  Electronics 6  Sport  6  Basis of programming  6 |

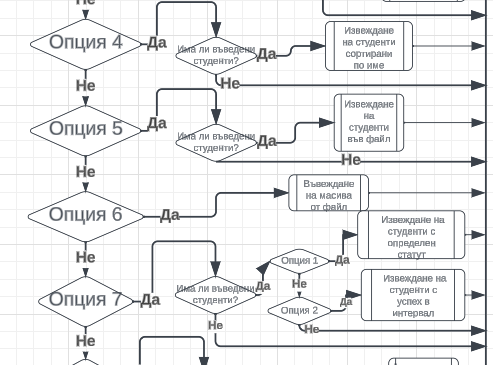
Реализация на условие A

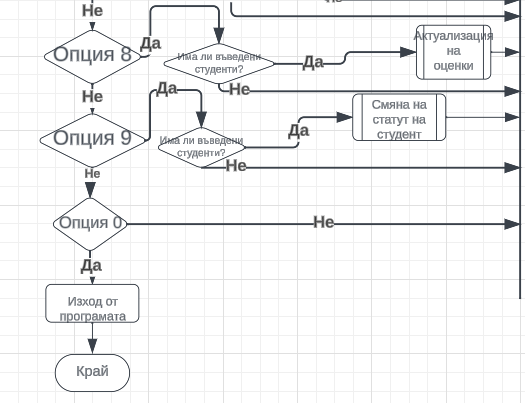
Меню с избор на функциите от програмата

В алгоритъма се използва switch, case statement като от потребителя се въвежда опция от 0 до 9. Първоначално се извеждат функционалностите на програмата. При въвеждане на опция въведения вход се валидира и ако е се различава от очаквания, на потребителя се казва да въведе валидна опция.Функцията int menu се извиква от главната функция int main().

Блок схема на алгоритъма







Функция, с която е реализиран алгоритъма

Алгоритъма е реализиран с функция int menu

#### Входни данни на функцията

Опция от 0 до 9. Като 0 е за изход от програмата. Всякаква друга опция е невалидна и това се пояснява на потребителя.

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

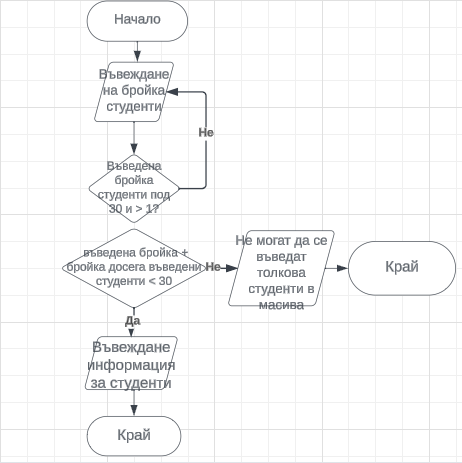
При невалиден избор на опция се пояснява на потребителя, че е въвел невалидна опция.

Реализация на условие B

Алгоритъм въвеждане на студенти в масива

Програмата пита потребителя колко студенти иска да въведе в масива. Ако въведе число е по-малко от 1 или по-голямо от 30 се извежда грешка която казва на потребителя да въведе валиден брой студенти. Ако въведените досега студенти са по-малко от 30 на потребителя се позволява да въведе студенти, ако те са 30 се извежда пояснително съобщение което казва на потребителя, че не могат да се добавят повече студенти в масива.

Блок схема на алгоритъма



Функция, с която е реализиран алгоритъма

Алгоритъма е реализиран с функция int input. Изполвана е помощна функция return\_num\_students за броене на досега въведените студенти

#### Входни данни на функцията

Бройка студенти

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

Функцията връща 0 при успешно изпълнение. Ако въведе число по-малко от 1 или по-голямо от 30 се извежда грешка която казва на потребителя да въведе валиден брой студенти. Ако са въведени максималния брой студенти на потребителя се извежда пояснително съобщение което му казва, че не могат да се въведат повече студенти.

Помощни функции

Изполвана е помощна функция return\_num\_students за броене на досега въведените студенти

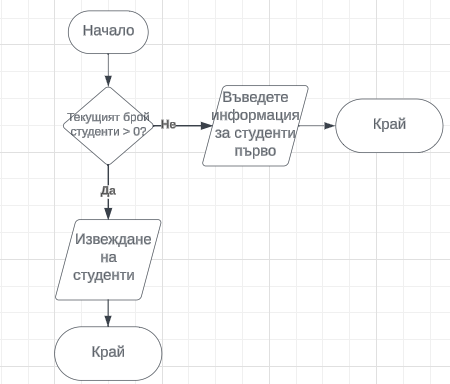
Реализация на условие C

Алгоритъм извеждане на студенти от масива

Извеждат се всички досега въведени студенти. Ако няма въведени студенти досега се извежда пояснително съобщение казващо, че няма въведени студенти в масива досега и няма какво да се извежда.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.



Функция, с която е реализиран алгоритъма

int output. Изполвана е помощна функция return\_num\_students за установяване на бройката на досега въведените студенти

#### Входни данни на функцията

Функцията няма входни данни

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

Ако няма въведени студенти в масива при избиране на опцията се извежда пояснително съобщение казващо на потребителя, че няма въведени студенти в масива.

Помощни функции

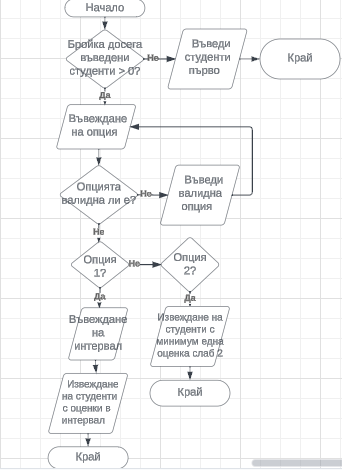
Изполвана е помощна функция return\_num\_students за броене на досега въведените студенти

Реализация на условие D

Алгоритъм търсене и извеждане на екран

В алгоритъма се използва функция с подменю което пита потребителя дали иска да търси и извежда студенти с оценка по БП в интервал или дали иска да търси студенти с поне една оценка слаб 2.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Алгоритъма се реализира чрез функция int search\_in\_range\_or\_look\_for\_fail

Изполвана е помощна функция return\_num\_students за установяване на бройката на досега въведените студенти

#### Входни данни на функцията

Въвежда се опция 1 или 2. При избор на всякакво друго число се извежда съобщение грешка. Ако се избере опция 1 се въвежда долната граница на интервала и горната граница. Долната граница не може да е по-малка от 2 ,а горната не може да е по-голяма от 6. Ако потребителя въведе по-малко от 2 или по-голямо от 6 се извежда пояснително съобщение. Ако се избере опция 2 се извеждат студенти с минимум една оценка Слаб 2. Ако няма студенти с оценка Слаб 2 се извежда съобщение което пояснява това на потребителя

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

При опция 1 се извеждат имената на студенти с оценка по БП в интервала, а в опция 2 се извеждат имената на студенти, който имат минимум една оценка Слаб 2.

Помощни функции

Изполвана е помощна функция return\_num\_students за броене на досега въведените студенти

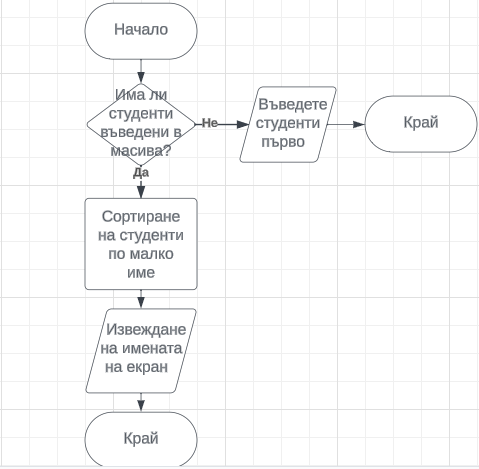
Реализация на условие E

Сортиране и извеждане на студенти на екран

В алгоритъма се сортират студенти по малко име и се извеждат целите имена на студенти.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.



Функция с която е реализиран алгоритъма

Алгоритъма се реализира с функция int sort\_by\_firstname

#### Входни данни на функцията

Функцията няма входни данни. За да се стартира се избира опция 4.

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

Извежда се съобщение “Sorted Students by First Name” и се извеждат студентите сортирани по малко име.

Помощни функции

Изполвана е помощна функция return\_num\_students за броене на досега въведените студенти

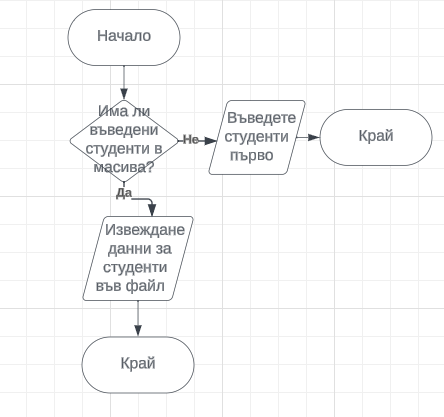
Реализация на условие F

Работа с външен файл

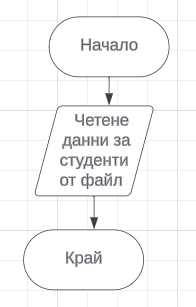
Масива от студенти се записва във външен файл и се въвежда(чете) от външен файл

Блок схема на алгоритъма

Блок схема за извеждане



Блок схема на функция за четене от файл



Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с две функции int write\_to\_file и функция int read\_array\_from\_file

#### Входни данни на функцията

Няма входни данни. Функцията за извеждане във файл се избира със опция 5 и функцията за четене от файл се избира с опция 6.

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

Пояснително съобщение което казва, че четенето или записването във файла е успешно.

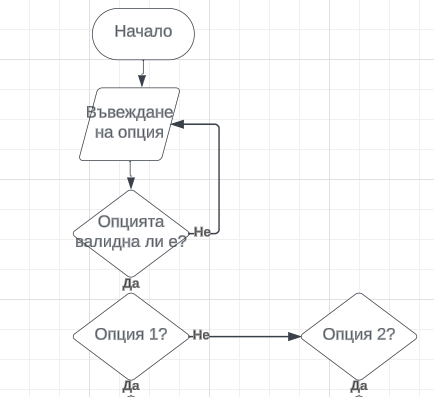
Реализация на условие G - допълнение първо

Създаване на подменю към основното с функции за отделяне на студенти по статут и сортиране по факултетен номер, и отделяне на студенти по успех в интервал и сортиране по ЕГН

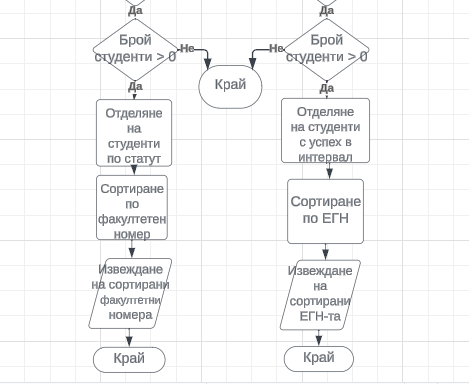
Създадена е функция в която се съдържа меню за избор между функцията за отделяне на студенти по статут и отделяне на студенти с успех в интервал. Ако се избере невалидна опция се извежда съобщение грешка и потребителят трябва да въведе опция пак.

Блок схема на алгоритъма

Част 1 блок схема избиране на опция



Част 2 на блок схема



Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с функция int menu\_for\_function\_seperate\_students\_by\_grades\_or\_status() , като в нея се съдържа меню за избор на една от функциите int seperate\_students\_by\_status() и int separate\_students\_by\_grades(). Използва се и функция return\_num\_students

#### Входни данни на функцията

Въвежда се опция 1 за отделяне на студенти по статут и опция 2 за отделяне на студенти с успех в интервал.

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

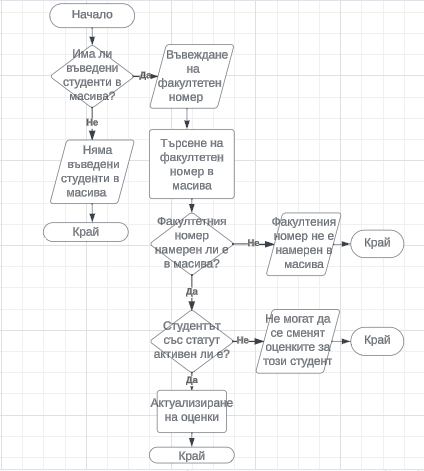
Извеждат се отделените студенти по статут или по оценки. Ако се въведе невалидна опция( различна от 1 или 2) се извежда пояснително съобщение, че е въведена невалидна опция.

Реализация на условие H - допълнение второ

Въвеждане/Актуализация на оценки и среден успех за студент

Функцията позволява актуализация на оценки като съответно средния успех се смята наново. Актуализацията става по въведен факултетен номер като програмата търси съвпадение. Ако няма съвпадение се извежда съобщение гласящо, че студента го няма в базата данни. Ако няма въведени студенти се извежда съобщение, че няма въведени студенти. Проверява се статута на студента и ако е различен от “active” се извежда съобщение което казва на потребителя, че не могат да се актуализират оценките за този студент.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Използва се функция int change\_data. След промяна оценките се смята средния успех наново. Използва се и функция int calculate\_average\_grade за смятане наново средния успех и също така string convert\_to\_lower\_case за изпълняване на програмата правилно дори ако отговорите “yes” или “no” са въведени с големи букви.

#### Входни данни на функцията

Въвежда се факултен номер на студент. След това се за всеки предмет потребителя се пита дали иска да направи промени. Ако напише “yes” тогава му се дава опцията да въведе оценката за съответния предмет. Ако напише “no” тогава се преминава към следващия предмет.

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

Програмата връща 0 при правилно изпълнение, а при различните видове грешки връща пояснителен текст. При въвеждането също се извежда пояснителен текст какво трябва да се въведе.

Реализация на условие I

Смяна статут на студент

Въвежда се факултетен номер. Ако се намери студент с въведения факултетен номер се въвежда и новия статут. При текущ статут активен или прекъснал се позволява промяна, а при текущ статут завършил не се позволява промяна.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Използва се функция int change\_data и се изполват помощни функции return\_num\_students, и convert\_to\_lower\_case

#### Входни данни на функцията

Въвежда се факултетен номер, ако статута на студента е “active” или “dropout” се въвежда нов статут на студент.

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

Извеждат се съобщения за грешки и пояснителни съобщения за въвеждането.

Реализация на условие J - допълнение трето

Попълване на данни автоматично от файл и записване във файл при затваряне на програмата

Изполват се две функции две функции int write\_to\_file и функция int read\_array\_from\_file

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с две функции int write\_to\_file и функция int read\_array\_from\_file като те са поставени на подходящите места.

#### Входни данни на функцията

Функцията чете автоматично данни за студентите от файл при стартиране на програмата.

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

Извеждат се пояснителни съобщение, че данните са съответно успешно въведени от файла при стартиране на програмата и, че са записани при затваряне.

Упътване за употреба

Условие A

При изпълнението на програмата на потребителя се извеждат всички възможни функции на програмата. Потребителят може да избере опция от 0 до 9, като 0 е резервирано за излизане от програмата. При изпълнение на програмата се зареждат автоматично примерни входни данни и при затваряне на програмата те се запазват.

Условие B

Чрез избиране на опция 1 от менюто се дава възможност на потребителя да въведе студенти собственоръчно. При избирането на опцията потребителя трябва да въведе колко студенти иска да добави към масива.

Условие C

При избиране на опция 2 на екрана се извеждат всички студенти в табличен вид. Ако няма студенти се извежда съобщение гласящо, че няма студенти в масива.

Условие D

При избор на опция 3 на потребителя се дават две опции. Ако потребителя въведе опция 1 се избира да се търсят студенти с оценка по БП в интервал като потребителя трябва да въведе този интервал. След търсенето ако са намерени студенти се извеждат на екрана. Ако потребителя въведе опция 2 се търсят студенти с минимум една оценка слаб 2.

Условие E.

При избор на опция 4 студентите се сортират по малко име по азбучен ред и се извеждат на екрана.

Условие F

При избор на опция 5 данните са студентите се извеждат в текстови файл, а при избор на опция 6 данните за студентите се въвеждат от текстови файл.

Допълнение първо – Условие G

При избор на опция 7 има създадено подменю към основното. На потребителя се дават две опции 1 или 2.

1. При въвеждане на опция 1 на потребителят се извеждат валидните статуси който студенитите могат да имат. Потребителя трябва да въведе един от тези статути. След въвеждането му се извеждат сортирани факултетните имена на студентите с този статут.
2. При избор на опция 2 потребителят трябва да въведе интервалът за средна оценка, който се използва за отделяне на студентите. Потребителят трябва да въведе две дробни числа като първото е долната граница за среден успех а второто е горната граница за среден успех.

Допълнение второ – Условие H

1. При избор на опция 8 потребителят трябва да въведе факултетнят номер на студента който ще бъде търсен. След въвеждане на факултетен номер ако студентът е намерен в масива потребителят е питан дали иска да промени оценката за всеки един от предметите който студента има. Ако напише “y es” тогава той трябва да въведе новата оценка за съответния предмет. Ако напише “no” тогава се преминава към следващия предмет. Ако студентът няма въведени оценки преди избиране на опция 8, не се пита за потвърждение преди въвеждане на всяка оценка. Опция 8 позволява промяна на оценки само когато студентът с въведения факултетен номер е със статут активен.
2. При избор на опция 9 потребителят трябва да въведе факултетният номер на студента на който иска да смени статутът. Ако студентът го има в масива и е със статут активен или прекъснал се допуска промяна на статуът, иначе при статут завършил не се допуска промяна.

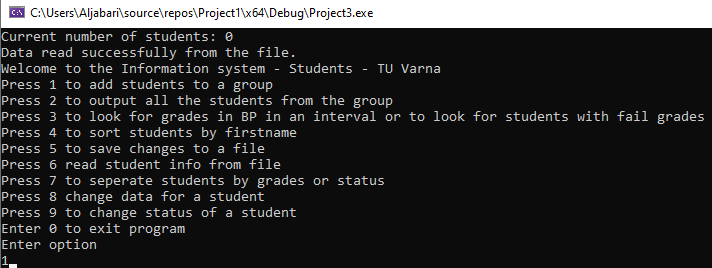
Допълнение трето – Условие J

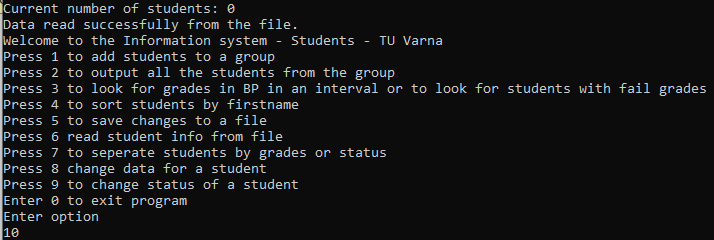
При пускане на програмата данните за студентите се попълват автоматично от текстови файл и при затваряне на програмата те автоматично се записват в текстови файл

Примерно действие на програмата

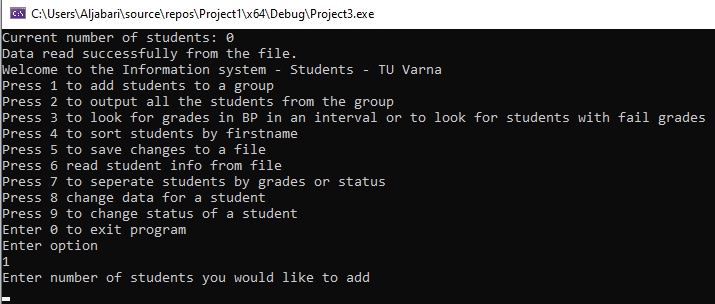
Условие A

Снимка на изгледа с примерни входни данни

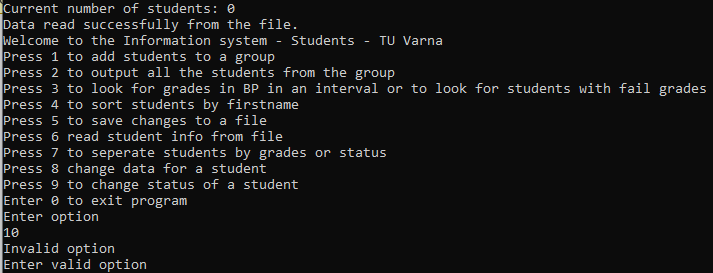




Снимка на изгледа с примерни изходни данни

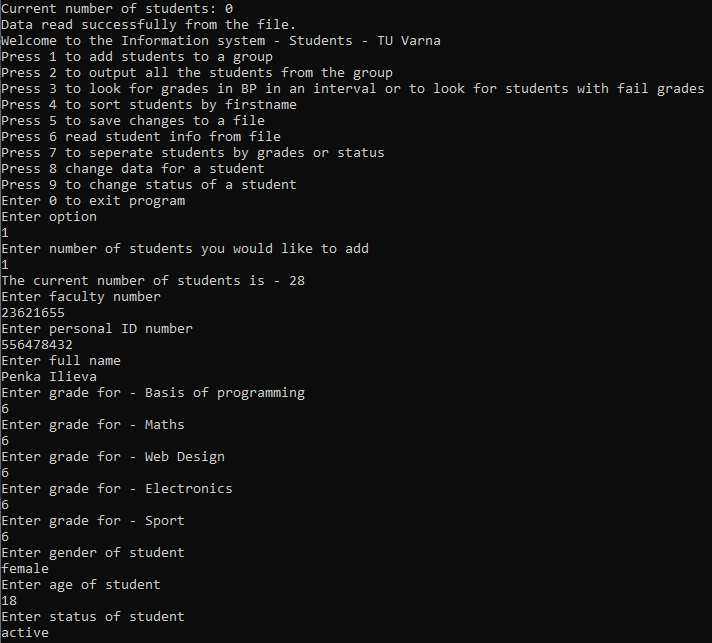


При въвеждане на опция различна от дадените се извежда съобщение което казва на потребителя, че въведената опция е невалидна и потребителят трябва да въведе нова валидна опция.

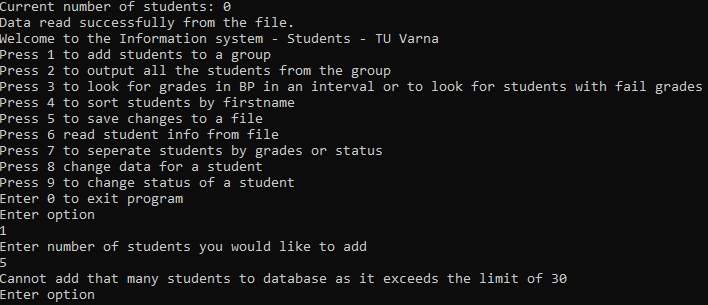


Условие B

Снимка на изгледа с примерни входни данни



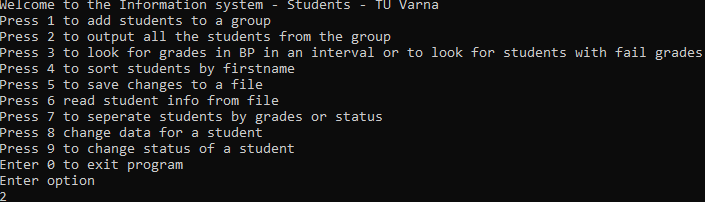
Снимка на изгледа с примерни изходни данни

Ако се опитаме да надхвърлим лимита от 30 студенти в група се извежда съобщение грешка. В случая има 29 студенти в масива и потребителя се опитва да въведе още 5 студента, но програмата не му позволява.

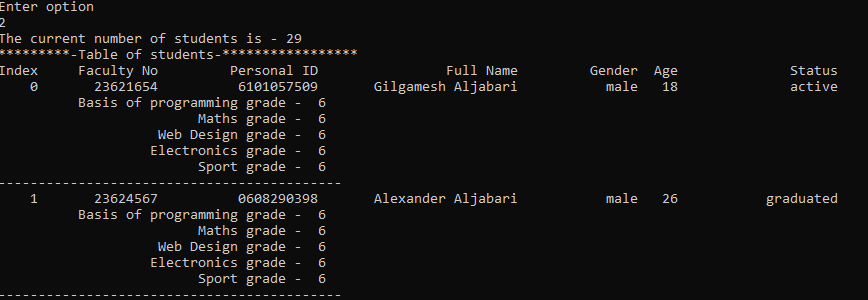
Условие C

Снимка на изгледа с примерни входни данни

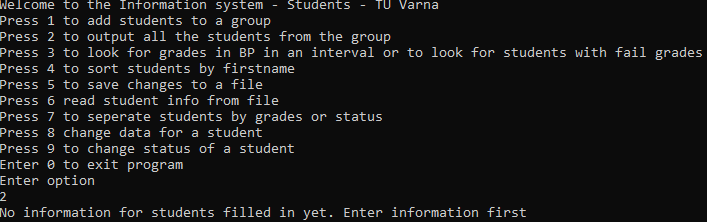
Въвеждане на опция 2 за извеждане на студенти в таблица



Снимка на изгледа с примерни изходни данни



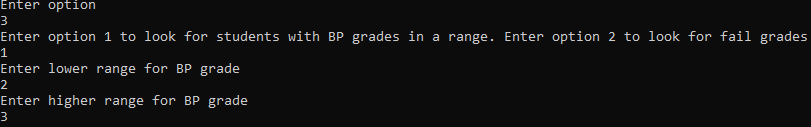
Ако няма въведени студенти в масива се извежда пояснително съобщение



Условие D

Снимка на изгледа с примерни входни данни

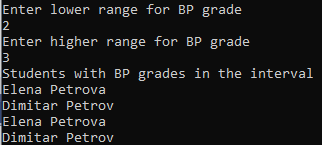
При опция 1 се въвеждат следните данни



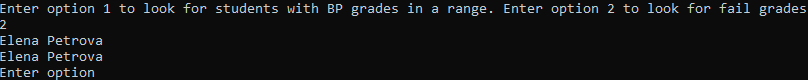
При избиране на опция 2, в подменюто не се въвеждат данни.



Снимка на изгледа с примерни изходни данни



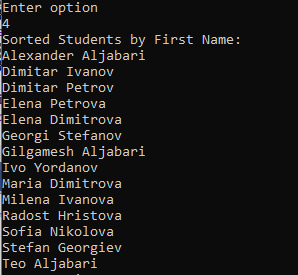
При опция 2 се извеждат студентите с поне една оценка слаб.



Условие E

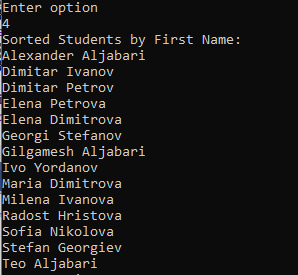
Снимка на изгледа с примерни входни данни

Избиране на опция 4 за сортиране на студенти по малко име



Снимка на изгледа с примерни изходни данни

Извеждане на студенти сортирани по малко име по азбучен ред



Условие F

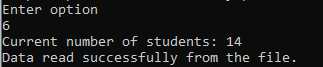
Снимка на изгледа с примерни входни данни

Избира се опция 5 – за запазване на масива във файл и опция 6 за четене от файл.

Снимка на изгледа с примерни изходни данни

При избор на опция 5 и успешно запазване на масива от студенти във файл се изписва съобщение което казва на потребителя, че успешно са запазени данните 

При избор на опция 6 и успешно четене се изписва съобщение което гласи, че четенето е успешно. Също така се изписва текущия брой на студенти.

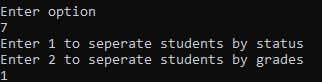


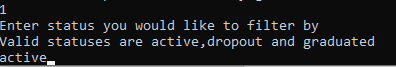
Допълнение първо – Условие G

Снимка на изгледа с примерни входни данни

При въвеждане на опция 7 потребителят влиза в подменю в което трябва да се избере една от опциите 1 или 2. Опция 1 е за отделяне на студенти по статут а опция 2 е за отделяне на студенти по среден успех

При избор на опция 1 се въвежда един от статутие по който да се отделят студентите.



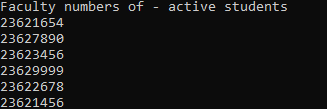


При избор на опция 2 се въвежда минумим и максимум за среден успех по който да се отделят студентите

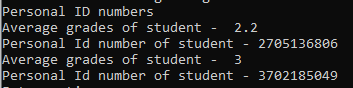


Снимка на изгледа с примерни изходни данни

При избор на опция 1 след въвеждане на статут се извеждат отделените студенти с въведения статут



При избор на опция 2 след въвеждане на минумум и максимум за средни оценки се извеждат отделените ЕГН-та със средните оценки на студентите.



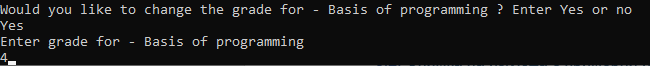
Допълнение второ – Условие H

Снимка на изгледа с примерни входни данни

Избира се опция 8 от менюто като при нейното избиране потребителят трябва да въведе факултетният номер на студента за който иска да прави промени по оценките



За всяка оценка на студента се пита дали потребителят иска да направи промяна ако потребителят потвърди тогава се въвежда нова оценка за съответния предмет



Снимка на изгледа с примерни изходни данни

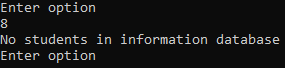
В случай, че студентът е със статут dropout или със статут graduated не се позволява промяна на оценки и се извежда съобщение грешка



Ако не бъде намерено съвпадение с въведения факулетен номер в масива се извежда съобщение казващо на потребителя, че не е намерен студент с такъв факултетн номер



Ако няма студенти в масива се извежда съобщение казващо на потребителя, че няма студенти в масива



Допълнение трето – Условие J

Снимка на изгледа с примерни входни данни

Въвежда се факултетеният номер на студента на който искаме да сменим статутът 

След това се въвежда новия статут на студентът

Снимка на изгледа с примерни изходни данни

Ако въведения факултетен номер на студента не е намерен в масива се извежда съобщение което казва на потребителя, че не е такъв факултетен номер в масива



Ако студентът с въведения факултетен номер е със статут завършил не се позволява промяна на статута и се извежда съобщение на потребителя поясняващо това



Ако няма въведени студенти в масива се извежда съобщение на потребителя което му казва, че няма въведени студенти в масива.



## **Помощни функции**

## 10.1 Помощна фунцкия 1

Функция int return\_num\_students – се използва за броене на досега въведени студенти. Използва се цикъл който да брой докато стигне студент на когото булевата променлива active не е true. Тази променлива става true когато има въведена информация за студента. Когато цикъла стигне променлива която не е true това означава, че няма въведена инфромацияза студента. В края на функцията стойноста до която е стигнал брояча се присвоява на променливата x която се връща от функцията. Функцията се изполва във почти всяка една от другите функции.

## 10.2 Помощна фунцкия 2

Функция string convert\_to\_lower\_case се изполва за превръщане на string от големи във малки букви. Изполва се for цикъл като всеки символ от result става от големи в малки букви.

## Помощна фунцкия 3

Функция int calculate\_average\_grade смята средния успех на студентите.

## Помощна фунцкия 4

Функция int swap е помощна функция за размяната местата на два елемента в масива. Използва се в сортировката на малки имена