Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „Компютърни науки и технологии“

**СЕМЕСТРИАЛНА ДОМАШНА РАБОТА**

по дисциплината „Базово програмиране”

на тема: „Информационна система Конкурс за красота”

Вариант 265

|  |  |
| --- | --- |
| Изготвил: Иван Георгиев Кръстев | Проверил: |
| Специалност: Изкуствен Интелект |  |
| Група: 1 |  |
| Факултетен номер: |  |

2024

Съдържание

[I. Задание на проекта 3](#_Toc186568535)

[II. Анализ на решението 4](#_Toc186568536)

[1. Структура за данните в програмата 4](#_Toc186568537)

[2. Реализация на условие A 4](#_Toc186568538)

[3. Реализация на условие B 5](#_Toc186568539)

[4. Реализация на условие C 7](#_Toc186568540)

[5. Реализация на условие D 8](#_Toc186568541)

[6. Реализация на условие E 10](#_Toc186568542)

[7. Реализация на условие F 11](#_Toc186568543)

[8. Реализация на условие G - допълнение първо 13](#_Toc186568544)

[9. Реализация на условие H - допълнение второ 15](#_Toc186568545)

[10. Реализация на условие I - допълнение второ 16](#_Toc186568546)

[11. Реализация на допълнение трето 18](#_Toc186568547)

[III. Упътване за употреба 20](#_Toc186568548)

[1. Условие A 20](#_Toc186568549)

[2. Условие B 20](#_Toc186568550)

[3. Условие C 21](#_Toc186568551)

[4. Условие D 21](#_Toc186568552)

[5. Условие E 21](#_Toc186568553)

[6. Условие F 22](#_Toc186568554)

[7. Допълнение първо 22](#_Toc186568555)

[8. Допълнение второ 23](#_Toc186568556)

[9. Допълнение трето 23](#_Toc186568557)

[IV. Примерно действие на програмата 23](#_Toc186568558)

[1. Условие A 23](#_Toc186568559)

[2. Условие B 24](#_Toc186568560)

[3. Условие C 24](#_Toc186568561)

[4. Условие D 25](#_Toc186568562)

[5. Условие E 26](#_Toc186568563)

[6. Условие F 27](#_Toc186568564)

[7. Допълнение първо 28](#_Toc186568565)

[8. Допълнение второ 30](#_Toc186568566)

[9. Допълнение трето 33](#_Toc186568567)

Задание на проекта

Да се напише компютърна програма, реализираща информационна система, която

поддържа конкурс за красота (номер на участник, име, възраст, пол, бедрена обиколка,

раменна обиколка, обиколка на шията, обиколка на прасеца). Програмата съхранява и

обработва данни за участниците в конкурса. Максималния брой участници е 30.

**Базова задача**

1. Меню за избор на функциите от програмата

**Функции от програмата са:**

1. Добавяне:

a. Добавянето трябва да позволява да се въвеждат различен брой нови участници като се допълват в масива, като не трябва да превишават максималния брой (30).

***Пример:***Добавяне на списък с участници. Въвежда се цяло число n, което позволява да се въведат n на брой участници. n не може да надвишава свободните елементи в масива.

1. Извеждане на екрана

a. Извеждане на всички участници в оформен вид

1. Търсене и извеждане на екрана:

a. Намиране и извеждане на всички участника с най-ниска възраст

b. Търсене на участник по име

1. Подреждане на основния масив, без да се извежда на екрана:

a. Подреждане на участниците по нарастващ ред на възрастта им

1. Управление на файл:

a. Извеждане на масива с участници във файл (двоичен)

b. Въвеждане на масива с участници от файл (двоичен)

**Допълнение първо (+ базова задача)**

1. Създайте подменю, в което се влиза от основното, с нови функции за:

а. Извеждане на участниците в подреден ред по години и име, без да се променя основния масив.

b. Търсене и извеждане на участниците по въведена възраст и пол

**Допълнение второ (+ базова задача)**

1. Разделяне на участниците в категории (10-11 седмица)

a. Разпределят се участниците във възрастови категории I (14-16), II (17-19), III (20-22), IV (23-25), всяка категория в отделен нов масив.

b. Извеждат се списъците на участниците от всяка категория, ако в дадена

категория няма участници се извежда съобщение за това;

1. Определяне на победителите от всяка категория: (9-10 седмица)

a. Победител в конкурса се определя по формулата за пропорционални телесни мерки (П) – П = Б/(Ш+Н+Р) (бедрена обиколка (Б), раменна обиколка (Р), обиколка на шията (Ш), обиколка на прасеца (Н));

b. Участниците с П между 0,54 и 0,62 се подреждат по намаляващ ред, за да се определи победителя;

c. Участниците с П повече от 0,63 се нареждат в нарастващ ред, преди участниците с П по-малко от 0,53 който се подреждат в нарастващ ред;

**Допълнение трето (+ базова задача)**

1. 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

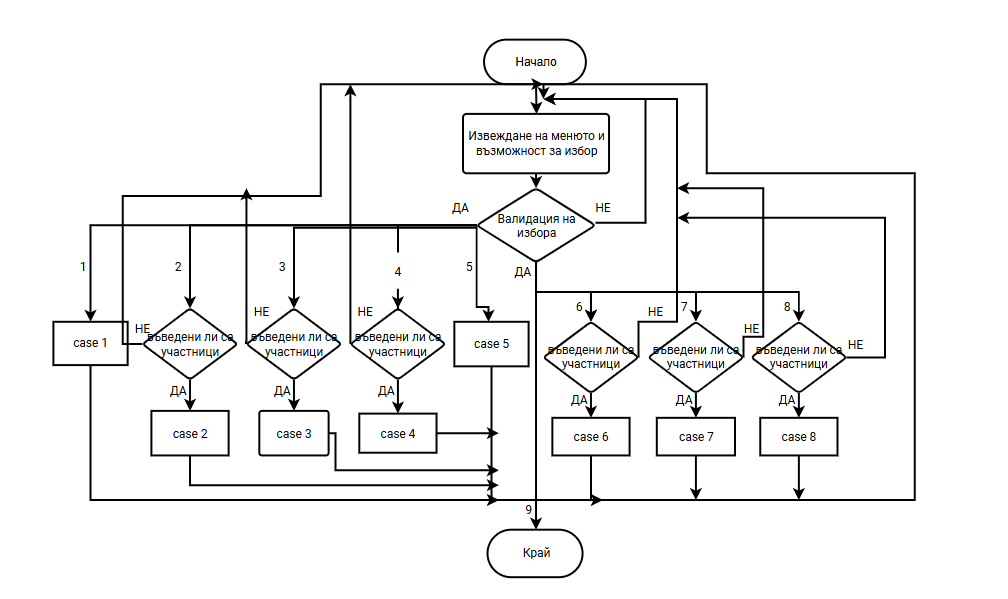
Структура за данните в програмата

|  |  |  |
| --- | --- | --- |
| Структура | Обяснение | Примерени стойности |
| struct Participants  {  int number, age;  char name[80];  char gender[7];  float hips, shoulders, necks, calves;  float results;  }; | * Number – ще съхранява номера на всеки участник * Age – ще съхранява възрастта на участника * Name – името на участника * Gender – полът на участника * Hips. Shoulders, Necks, Calves – съответно обиколката на бедрата, раменете, шията и прасците на участника * Results – резултата, който участникът получава по формула | 4  22  Mariya  Female  57.5  86.1  32.2  34.6  0.63 |

Реализация на условие A

Анализ на алгоритъма, който трябва да се реализира

Условие А изисква съставяне на меню на програмата. Със стартирането на програмата започва завъртане на цикъл, извеждайки меню с опции, от които потребителят може да избере. Дава се възможност да избере една от предоставените опции. Изборът му се валидира и при несъответствие се извежда съответно съобщение. В противен случай се изпълнява функцията, която е избрал, след което цикълът продължава. След всяка итерация конзолата се изчиства. При избор на точка 9, се напуска цикълът и програмата приключва.

Блок схема на алгоритъма

Функция, с която е реализиран алгоритъма

Int main()

Започва при стартиране на програмата и очаква коректен избор на точка от менюто, за да предприеме следващи действия.

#### Входни данни на функцията

Функцията няма входни данни.

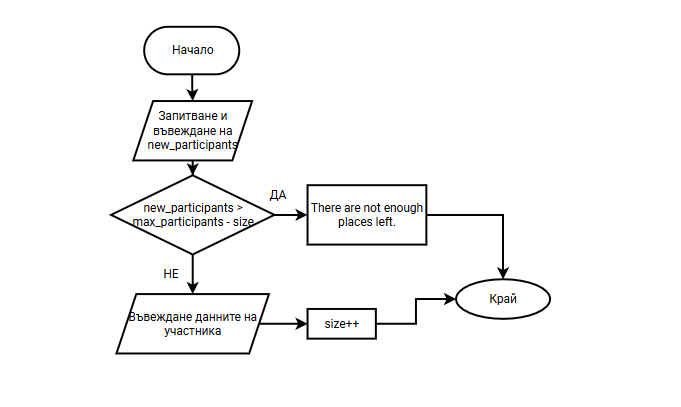
#### Изходни данни на функцията или данни, които се извеждат

На екрана се извеждат точките от менюто и се дава възможност на потребителя да избере една от тях.

Реализация на условие B

Анализ на алгоритъма, който трябва да се реализира

Условие B дава възможност за ръчно въвеждане на участиници в масива. Първоначално се пита за броя на новите участници, които потребителят би желал да добави. Прави се проверка дали този брой надвишава броя на оставащите места в масива. Ако това е така, се извежда съответното съобщение. В противен случай се дава възможност на потребителя да въведе данните на участника и се отбелязва увеличаването на масива с един участник.

Блок схема на алгоритъма

Функция, с която е реализиран алгоритъма

void Input(Participants participants[], int& size, int& max\_name, int &saved\_into\_file);

Функцията се извиква при желание на потребителя да въведе ръчно нови участници. Тази възможност се дава, ако броят на желаните нови участници е по-малък от оставащите места в конкурса.

#### Входни данни на функцията

Входните данни, които се подават на функцията са participants, size, max\_name. Participants е масивът с участници, чиито данни ще бъдат въвеждани. Извикваме size по адрес, тъй като тази променлива съхранява големината на масива и се използва при първоначалната поверка, а след това нейната стойност се инкрементира при всеки въведен участник. Променливата max\_name също се извиква по адрес, тъй като тя съхранява дължината на най-дългото име в масива, което ще ни трябва по-нататък. Прави се проверка на дължината на всяко въведено име и ако то е по-голямо, се присвоява новата стойност.

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

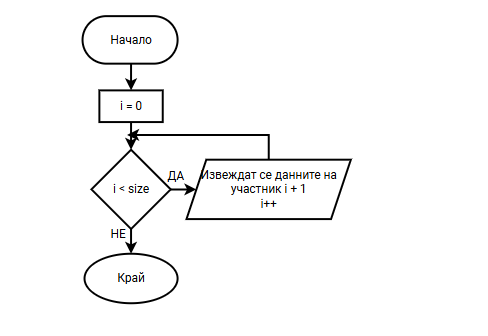
Изходните данни за функцията могат да бъдат size и max\_name, тъй като те се предават по адрес и тяхната стойност се променя.

Реализация на условие C

Анализ на алгоритъма, който трябва да се реализира

В условие C се изисква да бъдат изведени всички участници в конкурса. За целта се използва предварително изготвена таблица и се завърта цикъл, който извежда на екрана всички участници и техните данни, докато не свърши техния брой.

Блок схема на алгоритъма



Функция, с която е реализиран алгоритъма

void Output(Participants participants[], int size, int max\_name);

void HeaderRowOfTable(int max\_name);

void TitleRowOfTable(int max\_name);

void BodyOfTable(Participants participants[], int size, int max\_name, int i);

Функциите HeaderRowOfTable, TitleRowOfTable и BodyOfTable служат за предварителното оформяне на таблицата, в която ще бъдат изведени участниците. HeaderRowOfTable е заглавният ред, който се поставя в началото и в края на таблицата. TitleRowOfTable е редът оформящ колоните, в които ще бъдат изведени данните на участниците. BodyOfTable се използва за въвеждането на всеки участник форматирано. Output е основната функция на условието, като при извикване от менюто, тя отпечатва таблицата с всички участници и техните данни.

#### Входни данни на функцията

Като входни данни на функцията се подават масивът с участници participants, size и max\_name. Първите две служат за извеждането на участниците, а от max\_name имаме нужда, за да можем да го подадем на подфукнциите за таблица, тъй като тази променлива ще определи максималното заделено място за име в нея.

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

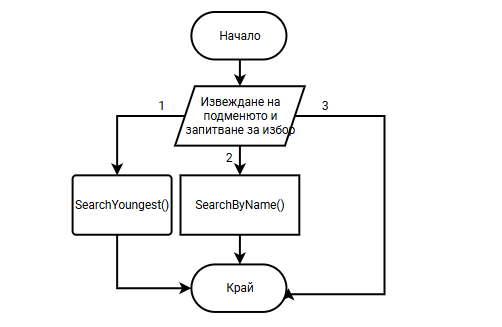
Функцията няма изходни данни, но извежда данните на всички участници на екрана във форматиран вид.

Реализация на условие D

Анализ на алгоритъма, който трябва да се реализира

Условието се изпълнява чрез подменю, което предлага възможност за намиране на участника с най-ниска възраст или търсене на участник по име.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

void SubMenu(Participants participants[], int size, int max\_name);

След извикване на функцията от главното меню се изчиства терминала и се предоставят нови подвъзможности за търсене на потребителя, от които той може да избере.

#### Входни данни на функцията

Входните данни на фукнцията са масивът participants, size и max\_name. Те ще бъдат предадени на подфункциите, които ще изпълняват възможните действия.

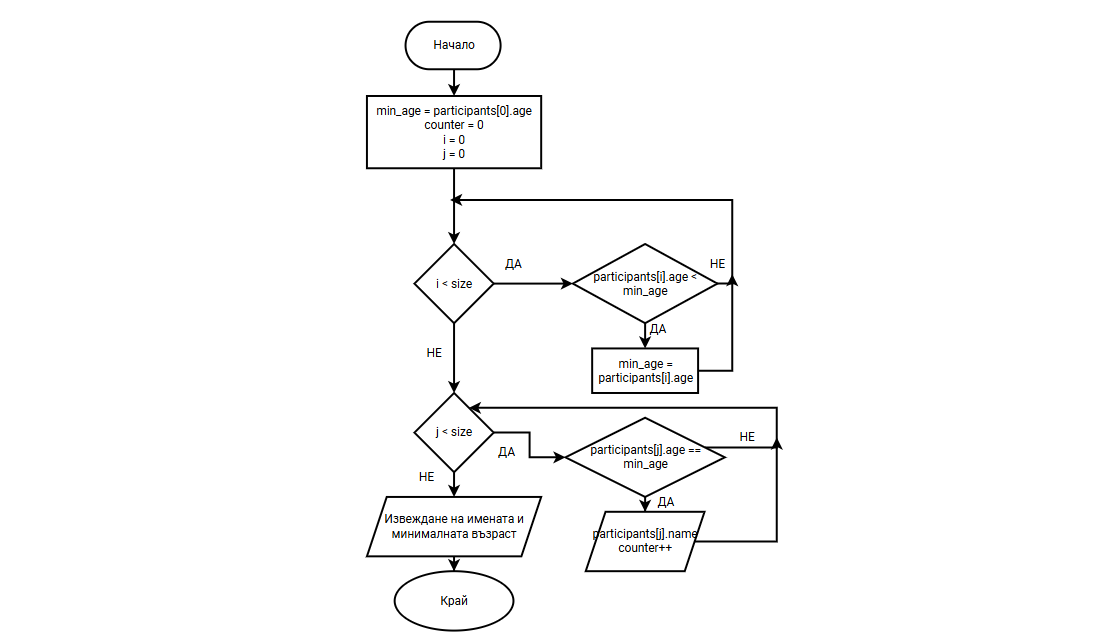
#### Изходни данни на функцията или данни, които се извеждат

Функцията няма изходни данни, а извежда на екрана подменю с възможности за избор.

### **A**. Анализ на алгоритъма, който трябва да се реализира

Създават се две променливи, едната ще следи минималната възраст на участник, като за първоначално се задава тази на първия участник в масива, а другата дали има повече от един участник на тази минимална възраст. Завърта се цикъл между всички участници, след който се установява най-ниската възраст. След това отново се преминава през участниците и се отпечатва името (или съответно имената, ако броячът показва 1 или повече) на участника с най-ниска възраст и се уточнява каква е тя.

**А.** Блок схема на алгоритъма



### **А.** Функция с която е реализиран алгоритъма

void SearchYoungest(Participants participants[], int size);

Функцията се извиква при избор на потребителя от подменюто да намери най-младия участник. Тя изпълнява това условие и връща резултата му.

#### **A.** Входни данни на функцията

За входни данни на функцията се подават масивът participants и неговия size. Те се използват, за да може да се премине през масива и да се отсеят участниците с най-ниска възраст.

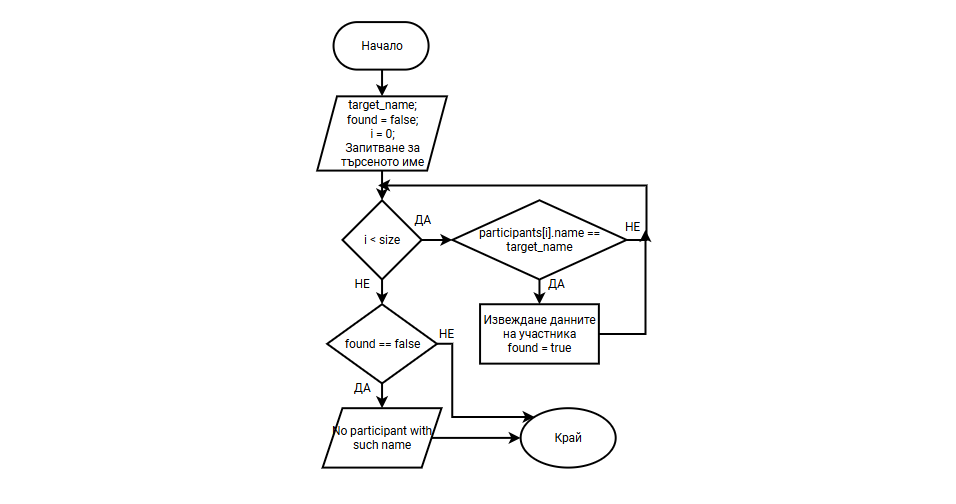
#### **А.** Изходни данни на функцията

Функцията няма изходни данни, но извежда на екрана имената на участниците с най-ниска възраст и съответната възраст.

### **Б**. Анализ на алгоритъма, който трябва да се реализира

Създават се две променливи, едната ще иска от потребителя търсеното име, а другата ще бъде флаг дали това име е намерено. След това се преминава през масива и се сравнява всяко име на участник дали съвпада с търсеното такова. При намиране на съответствие се извежда участника и данните му и се отбелязва, че има намерен участник с търсеното име. Цикълът обаче продължава, за да се провери дали има друг участник със същото име. Ако накрая не е открито съвпадение, се извежда съответното съобщение.

### **Б.** Блок схема на алгоритъма



### **Б.** Функция с която е реализиран алгоритъма

void SearchByName(Participants participants[], int size, int max\_name);

Функцията се извиква при желание на потребителя да търси участници по име след избор от подменюто. При извикването й се изисква търсено име, което се проверява дали съвпада с някое от тези в масива.

#### **Б.** Входни данни на функцията

Входните данни, които се подават на функцията са масивът с участници participants и size, които служат за преминаването през дължината на масива за проверка на съвпадение в търсеното име. Променливата max\_name се подава на функциите за таблицата.

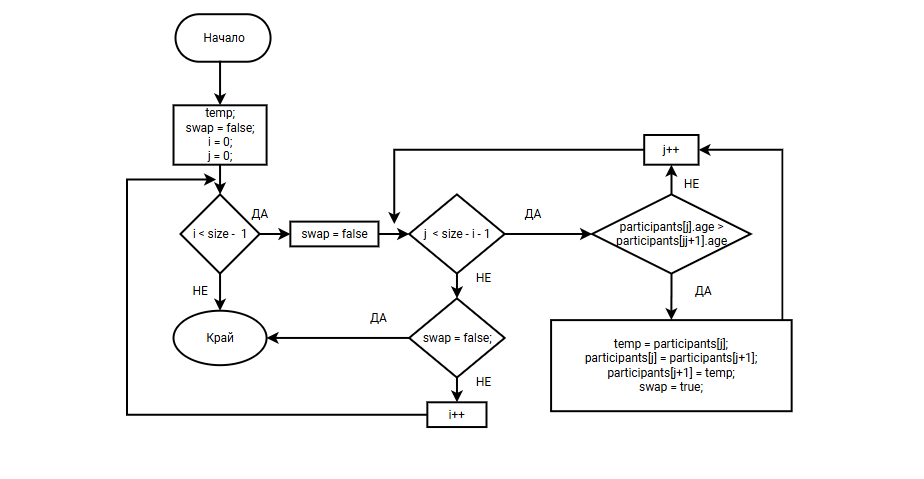
#### **Б.** Изходни данни на функцията

Функцията няма изходни данни, но извежда участник при съвпадение с търсеното име.

Реализация на условие E

Анализ на алгоритъма, който трябва да се реализира

За изпълнението на условие Е ще използваме сортиране чрез така наречения „метод на мехурчето“. Инициализират се две променливи, temp от типа на глобалната структура Participants, която ще съхранява данните на един от участниците при размяна, и булевата променлива swap, която следи дали при цяла итерация на външния цикъл ще има промяна в реда на елементите. Ако такава промяна не е намерена, сортирането приключва, тъй като очевидно няма да има повече промени.

Блок схема на алгоритъма

Функция с която е реализиран алгоритъма

void SortByAge(Participants participants[], int size);

Функцията се извиква след съответния избор от главното меню. Тя сортрира участниците по възраст във възходящ ред, без да извежда подредения масив на екрана.

#### Входни данни на функцията

Входните данни за функцията са масивът participants и size, които служат, за да могат да преминат циклите през цялостната дължина на масива и да сортират в съответния ред елементите.

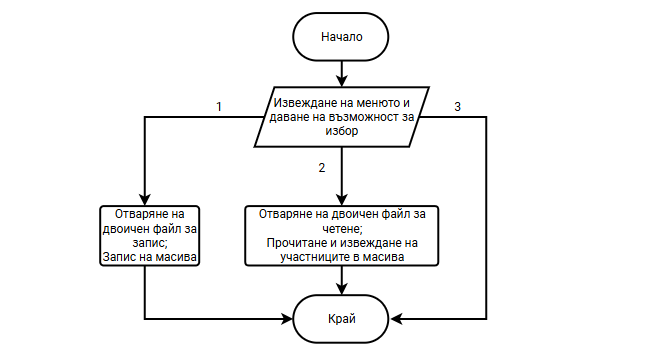
#### Изходни данни на функцията или данни, които се извеждат

Функцията няма изходни данни и не извежда нищо на екрана. Тя единствено подрежда елемнтите в масива по зададения критерий.

Реализация на условие F

Анализ на алгоритъма, който трябва да се реализира

За изпълнението на условие F се извежда подменю с възможност за потребителя да избере дали иска записване на данните във файл или четенето им от такъв. При избор на записване на масива във файл, той се записва двоично в него, а при избор на четене – съответно се въвежда масива в програмата отново от двоичен файл.

Блок схема на алгоритъма

Функция с която е реализиран алгоритъма

void FileSubMenu(Participants participants[], int &size, bool &entered, int &max\_name, int &saved\_into\_file);

void OutputFile(Participants participants[], int size);

void InputFile(Participants participants[], int &size, bool &entered, int &max\_name, int &saved\_into\_file);

Функцията FileSubMenu извежда менюто, от което потребителят може да избере четене или запис. При избор на запис в двоичен файл се извиква функцията OutputFile, която отваря файла изпълнява съответното действие и го затваря. При избор на четен от двоичен файл се извиква InputFile, отваря се двоичния файл, прочитат се данните и се извеждат на екрана, след което файлът се затваря..

#### Входни данни на функцията

Входните данни за функциите са масивът participants и size, които служат за преминаване през масива при записване и за попълването му при четене. На подменюто се подава и булевата променлива entered\_participants, за да се направи проверка при желание на потребителя за запис във файл дали има вече въведени участници. В противен случай се извжда съответното съобщение.

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

При четене от двоичния файл, променливата size се подава по адрес, тъй като тя се инкрементира с всеки добавен участник от файла. Отново при четенето се извеждат прочетените данни на терминала. При записването не се извежда нищо и програмата продължава работата си.

Реализация на условие G - допълнение първо

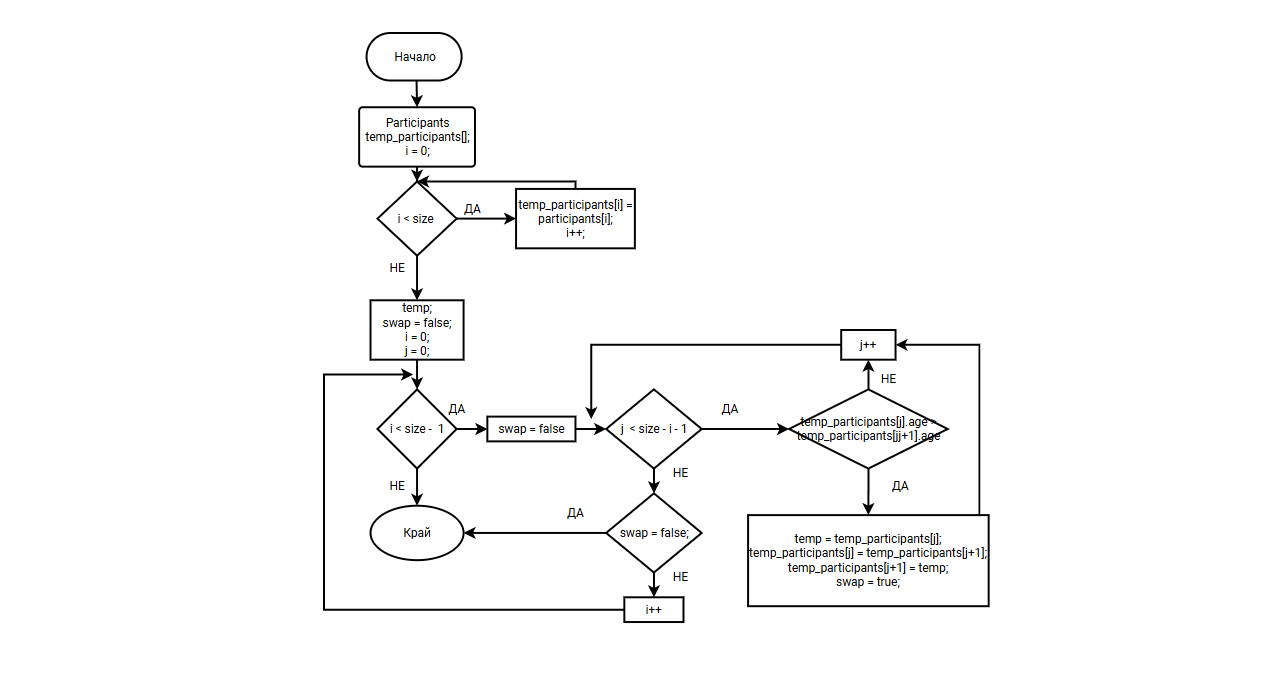
Анализ на алгоритъма, който трябва да се реализира

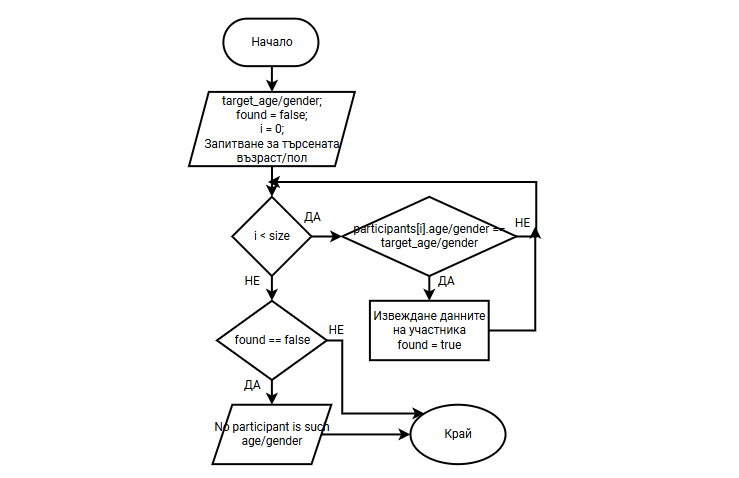
За реализацията на условие G се създава ново подменю с възможност да се изведат участниците в подреден ред по години и име, без да се променя основния масив и да се търсят и извеждат участници по въведена възраст или пол.

А) Инициализира се масив от структури, който ще бъде използван само във функцията, за да не се работи с основния и ще служи като негово копие. Затова чрез цикъл се записва целия основен масив в новия такъв. След това този нов масив се сортира във възходящ ред по години и накрая се извежда на екрана вече сортиран.

Б) По аналогичен начин се се правят две търсения – едното по възраст, другото по пол. Изисква се от потребителя търсения критерий и се преминава през масива в търсене на съответствия. При намерено такова, то се извежда на екрана, а в противен случай се извежда съответното съобщение.

Блок схема на алгоритъма

А) 

Б) 

Функция с която е реализиран алгоритъма

void AdditionalFunctions(Participants participants[], int size, int max\_name);

А) void SortByAgeWithoutChanging(Participants participants[], int size, int max\_name);

Функцията се извиква при желание на потрежителя да види участниците подредени във възходящ ред спрямо възрастта им. Тя подрежда участниците и ги извежда на екрана по желания критерий.

Б) void SearchByAge(Participants participants[], int size, int max\_name);

void SearchByGender(Participants participants[], int size, int max\_name);

Всяка от функциите се извиква при избор на потребителя от подменюто. Те извършват търсенето по години или пол и извеждат на екрана участниците, съвпадащи на тези критерии. Ако участник не е намерен по зададеното търсене, се изписва съответното съобщение.

#### Входни данни на функцията

А) Входните данни за функцията са масивът participants, size и max\_name. Първите две служат за презаписване на основния масив във временен такъв, а max\_name служи за форматиране при извеждане на таблицата.

Б) На функциите за търсене също се подават масивът participants и size, които се използват за преминаване през дължината му в търсене на съответствия с искания параметър, и max\_name, който е за форматиране на таблицата.

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

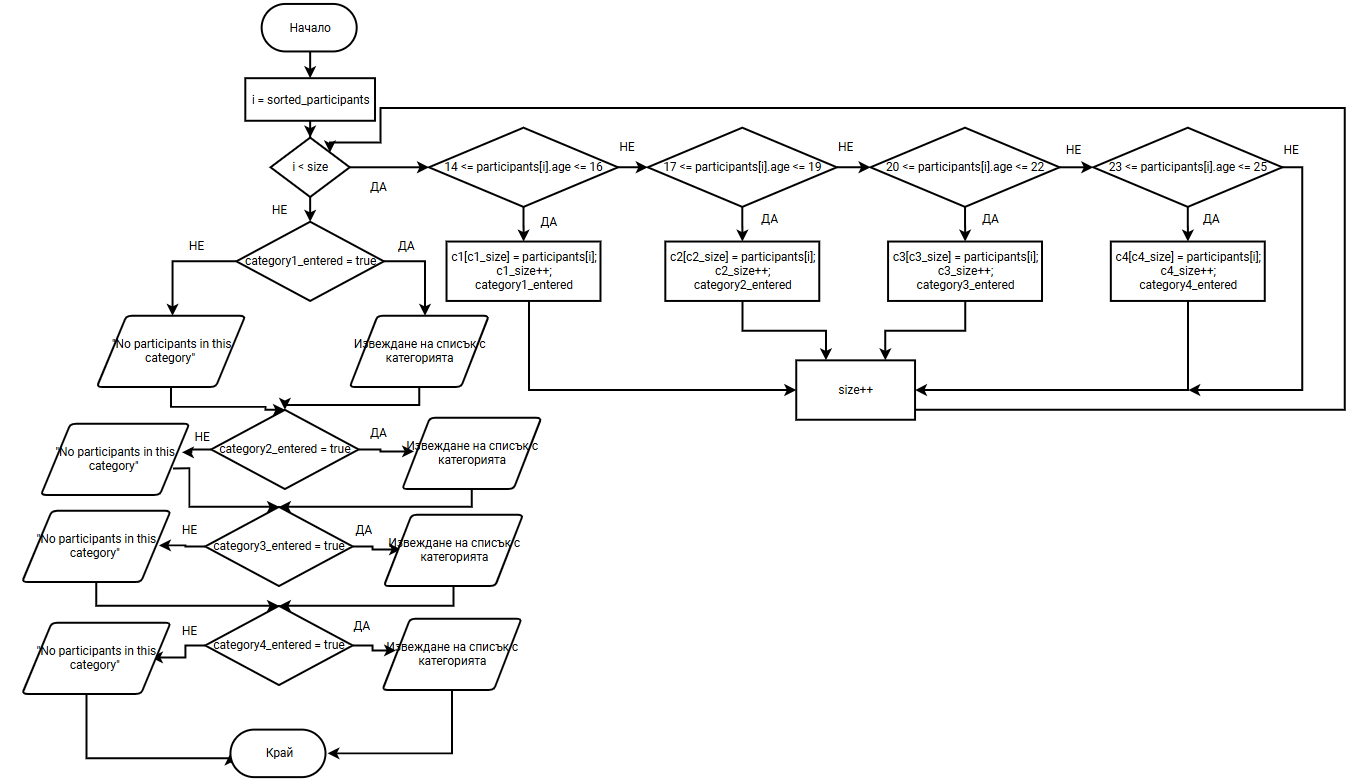
А) На екрана се извеждат участниците подредени в таблица във възходящ ред спрямо тяхната възраст.

Б) При намиране на съответствие се извеждат на екрана всички участници по търсения параметър.

Реализация на условие H - допълнение второ

Анализ на алгоритъма, който трябва да се реализира

За разпределянето на участници по категории се преминава през масива и се проверява всеки участник дали попада в някоя от категориите. Ако годините му влизат в категория, данните му се записват като елемент в масива на категорията. Накрая се извеждат на екрана списъци с категориите. Ако някоя от тях е празна, на мястото на списъка й се изписва съответното съобщение.

Блок схема на алгоритъма 

Функция с която е реализиран алгоритъма

void AgeCategories(Participants participants[], int size, Participants c1[], int& c1\_size, Participants c2[], int& c2\_size, Participants c3[], int& c3\_size, Participants c4[], int& c4\_size, int max\_name, int &sorted, bool &category1\_entered, bool& category2\_entered, bool& category3\_entered, bool& category4\_entered);

void SplitIntoCategories(Participants participants[], int size, Participants c1[], int& c1\_size, Participants c2[], int& c2\_size, Participants c3[], int& c3\_size, Participants c4[], int& c4\_size, bool& entered1, bool& entered2, bool& entered3, bool& entered4, int &sorted);

При избор на потребителя за това действие се извиква функцията AgeCategories. В нея първо SplitIntoCategories разпределя участниците по зададените категории и след това тези категории се извеждат на екрана.

#### Входни данни на функцията

На функцията се предават както основният масив participants и неговия размер size, така и масивите с категориите и техните размери и булевите променливи дали има въведени участници в тях и променливата sorted\_into\_categories, която следи колко от участниците вече са били разпределени.

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

Функцията променя масивите на категориите, както и техните размери и колко участника са разпределени при попълване, затова тези параметри се предават по адрес. Накрая на действието на функцията на екрана се извеждат списъци с категориите, ако в тях са въведени участници и съответно съобщение, ако не са.

## Реализация на условие I - допълнение второ

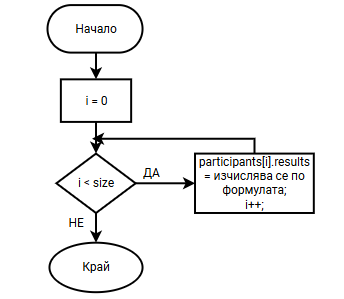
Анализ на алгоритъма, който трябва да се реализира

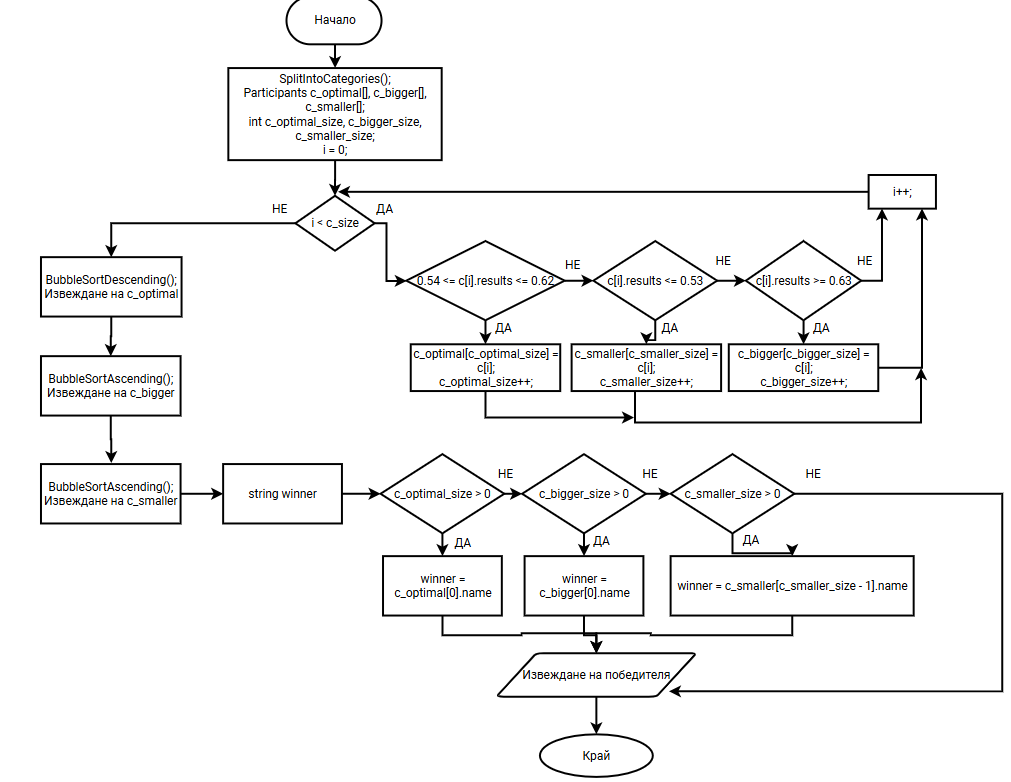
А) Използва се цикъл, който преминава през дължината на масива и за всеки участник изчислява резултатът му според зададената формула.

Б), В) Инициализират се нови масиви и променливи за дължините им, първият от които ще съдържа участниците с оценки в оптималния диапазон, вторият – тези с по-големи от 0.63 и третият – с по-малко от 0.53. Участниците се разпределят в тези категории, проверявайки резултата на всеки и поставяйки го в съответния масив, като увеличаваме и размера му. Впоследствие се подреджда масива с оптимално представилите се участници в намаляващ ред чрез „метода на мехурчето“, а след това останалите два – в нарастващ. След това се извежда целия готов списък на екрана. За победител се определя първият елемент в оптималния масив. Ако няма представители на оптималния масив, това ще бъде първия елемент в този с резултати, по-големи от 0.63, а ако и в него няма участници – последният от тези с резултати по-ниски от 0.53.

Блок схема на алгоритъма

А)



Б) 

Функция с която е реализиран алгоритъма

А) void CalculateResults(Participants participants[], int size);

Функцията се извиква, след като потребителят е избрал от основното меню да види победителите по категории и тя изчислява резултатие на участниците.

Б) void Winners(Participants p[], int size, int max\_name, Participants c1[], int &c1\_size, Participants c2[], int &c2\_size, Participants c3[], int &c3\_size, Participants c4[], int &c4\_size, int &sorted, bool& category1\_entered, bool& category2\_entered, bool& category3\_entered, bool& category4\_entered);

void CategoryWinner(Participants c[], int c\_size, int max\_name, int number\_category, int a, int b);

void BubbleSortDescending(Participants p[], int size);

void BubbleSortAscending(Participants p[], int size);

Основната функция се извиква след избор от основното меню. Тя извършва цялото действие от разпределянето на участниците по резултати, до определянето на победителя и извеждането на екрана на списъците за всяка една категория. За целта й помага функцията CategoryWinners, която пък вътре в себе си използва BubbleSortDescending и BubbleSortAscending за съответно подреждането в низходящ и възходящ ред, използвайки метода на мехурчето.

#### Входни данни на функцията

А) На функцията се предават основният масив participants и size, за да може да премине през него и да изчисли резултата на всеки участник.

Б) Функцията получава масивите за всички категории и техните размери, както и булевите променливи дали има въведени участници в тях. Масивите се използват за по-нататъшното разпределение на участниците в отделните категории, а булевите променливи – за да се направи проверка дали въобще са попълнени масивите.

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

А) Изходните данни за функцията са резултатите на всички участници, които се изчисляват и се запазват в масива от структури.

Б) Функцията няма изходни данни, но извежда на екрана подредените по резултати списъци за всяка категория и определя автоматично победителя, извеждайки неговото име.

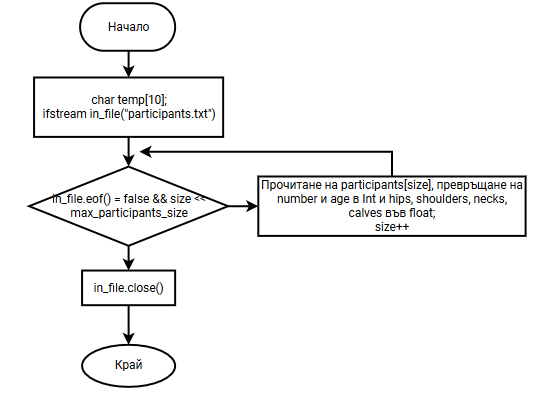
Реализация на допълнение трето

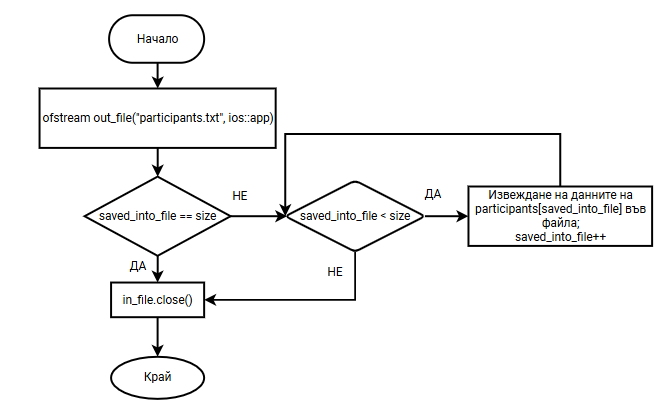
Анализ на алгоритъма, който трябва да се реализира

За автоматичното попълване на данните от файл се използва текстовият файл participants.txt. При стартиране на програмата, файлът се отваря, прочитат се данните от него, като числовите данни се конвертират към съответния си тип, зададен в структурата, попълват се основният масив и се затваря.

За автоматичното записване на данните във файл след затваряне на програмата отново се използва файлът participants.txt. Преди затваряне на програмата, той се отваря, проверява се дали има нововъведени участници и ако има – техните данни се попълват в края му и файлът се затваря.

Блок схема на алгоритъма





Функция с която е реализиран алгоритъма

void AutoInput(Participants participants[], int& size, int& max\_name, bool &entered, int &saved\_into\_file);

void AutoOutput(Participants participants[], int size, int &saved\_into\_file);

AutoInput се извиква всеки път при стартиране на програмата. Тя прочита данните от зададения й файл и попълва масива с тях, като ги превръща в съответния необходим тип.

AutoOutput се изиква както след избиране на потребителя да напусне програмата и да приключи действието й, така и при всеки въведен нов участник, за да се предотврати загуба на данни при непредвидени обстоятелства, и въвежда целия масив с участници в зададения файл.

#### Входни данни на функцията

Входни данни и за двете функции са масивът participants, size и променливата, която следи колко участници са били запазени във файл. Функцията за въвеждане има нужда от тях, за да попълни масива, а тази за извеждане – съответно да премине през него и да запише данните. На AutoInput се подават и max\_name и entered, тъй като тя може да ги измени в процеса на въвеждане на участниците.

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

Изходните данни за AutoInput са попълненият масив и променените size, entered, saved\_into\_file и евентуално max\_name.

Изходни данни за AutoOutput са изведените във файла данни на участници, както и променения брой на запазените във файла участници.

Упътване за употреба

Условие A

Очаквани входни данни

Условие А е изпълнено чрез менюто. На мястото за въвеждане на избор (Enter your choice) програмата очаква цяло число в диапазона между 1 и 9 като валиден вариант. При въвеждане на друго нещо, програмата извежда съобщение за това.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

При въвеждане на правилно число, програмата ще продължи, извиквайки функцията и изпълнявайки избраното действие.

Условие B

Очаквани входни данни

За въвеждането на участници в масива първоначално се изисква тяхната бройка, която трябва да бъде цяло число, ненадвишаващо максималния брой по условие (30). След това се въвежда номер, който отново трябва да е цяло число, име – текст, години – цяло число, обиколка на бедро, раменна обиколка, обиколка на шия и на прасец – реални числа. При несъответствие в типа на някоя от въведените данни се изписва съобщение.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Очакваният резултат е записванеето на нововъведените участници и техните данни в масива.

## Условие C

Очаквани входни данни

За да се изведе списък с всички участници, трябва да се избере 2 в основното меню.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

На екрана се извежда списък с всички участници и техните данни.

## Условие D

Очаквани входни данни

За да се отвори подменюто с възможност за търсене, от главното меню се избира 3. В подменюто, също като в главното, се въвежда подходящ избор цяло число в диапазона 1-3. При неправилен избор се извежда съобщение.

A) За да се намери най-младият участник, от подменюто се избира 1.

Б) За да се извърши търсене по име, от подменюто се избира 2. Изисква се въвеждане на търсеното име. Съвпадение ще има само ако името изцяло съвпада с това на въведен участник.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

А) На екрана ще се изведе името (или имената, ако са повече от един) на най-младия участник и неговата възраст.

Б) При намиране на участник със същото име като търсенето, то неговите данни ще се изведат на екрана. В противен случай се извежда съобщение, че няма участник с това име.

## Условие E

Очаквани входни данни

За да се подреди масивът във възходящ ред спрямо възрастта, от главното меню се избира 4.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Масивът се подрежда по желания критерий, но не се извежда на екрана.

## Условие F

Очаквани входни данни

За да се отвори подменюто с възможност за работа с двоичен файл, от главното меню се избира 5. В подменюто, също като в главното, се въвежда подходящ избор цяло число в диапазона 1-3. При неправилен избор се извежда съобщение.

А) За да се запишат участниците в двоичен файл, от менюто се избира 1.

Б) За да се прочетат от двоичен файл и да се попълни масивът, от менюто се избира 2.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

А) Участниците се записват в двоичен файл.

Б) Участниците се прочитат от двоични файл и масивът се попълва с данните им.

## Допълнение първо

Очаквани входни данни

За да се отвори подменюто с допълнителни функции, от главното меню се избира 6. В подменюто, също като в главното, се въвежда подходящ избор цяло число в диапазона 1-4. При неправилен избор се извежда съобщение.

А) За да се подредят участниците във възходящ ред по възраст, без да се променя основния масив, от менюто се избира 1.

Б) За да се извърши търсене по възраст, от менюто се избира 2. Трябва да се въведе възрастта, по която да се търси, като възрастта се очаква да бъде цяло число.

В) За да се извърши търсене по пол, от менюто се избира 3 и се въвежда полът, по който да се търси.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

А) Участниците се подреждат във възходящ ред по възраст и се извеждат на екрана във форматиран вид.

Б) На екрана се извеждат всички участници от търсената възраст и техните данни. В случай, че няма нито един участник на търсената възраст, се извежда съответното съобщение.

В) На екрана се извеждат всички участници от търсения пол и техните данни. В случай, че няма нито един участник от търсения пол, се извежда съответното съобщение.

## Допълнение второ

Очаквани входни данни

За да се разпределят участниците по категории и да се изведат списъците на екрана, от главното меню се избира 7.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

На екрана се извеждат списъци с участниците от всяка категория. Ако нито един участник не попада в някоя категория, се извежда съобщение за това.

## Допълнение трето

Очаквани входни данни

За да се подредят категориите по резултати и да се определят победителите във всяка от тях, от главното меню се избира 8.

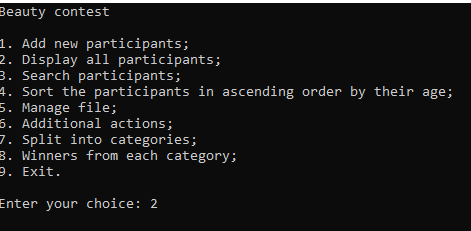
Очакван резултат от изпълнението на конкретния фрагмент от проекта

На екрана се извеждат подредени според заданието списъци по резултати и отдолу автоматично се определя победителя във всяка категория.

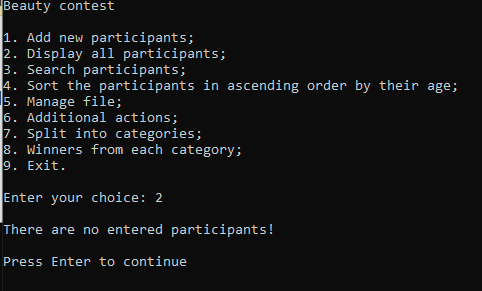
Примерно действие на програмата

Условие A

Снимка на изгледа с примерни входни данни

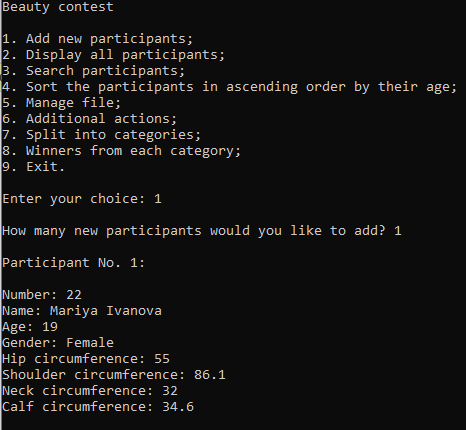


Снимка на изгледа с примерни изходни данни



Условие B

Снимка на изгледа с примерни входни данни

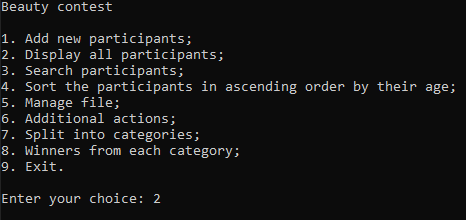


Снимка на изгледа с примерни изходни данни

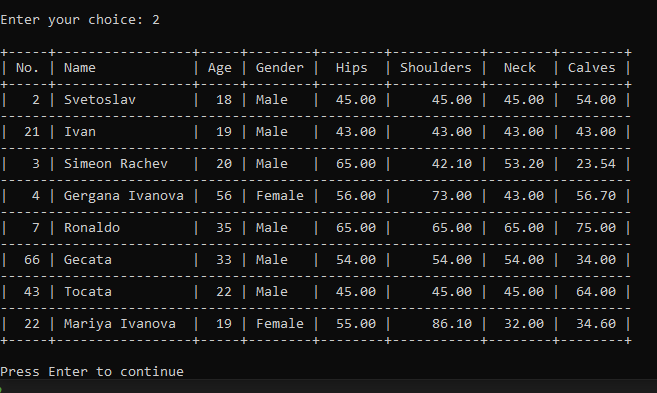
Данните се записват в масива и няма изходни данни.

Условие C

Снимка на изгледа с примерни входни данни

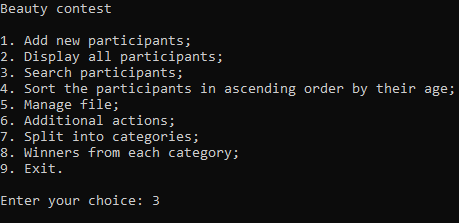


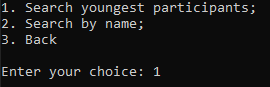
Снимка на изгледа с примерни изходни данни



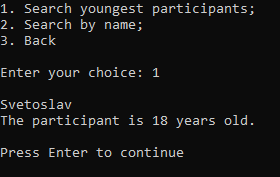
Условие D

Снимка на изгледа с примерни входни данни



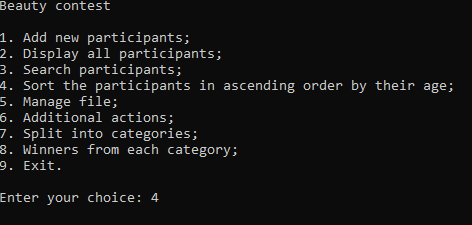


Снимка на изгледа с примерни изходни данни



Условие E

Снимка на изгледа с примерни входни данни

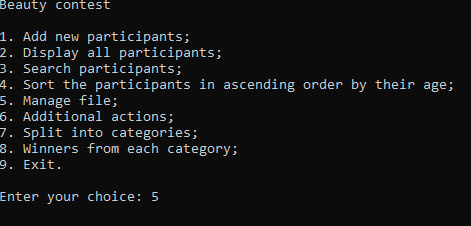


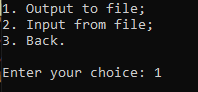
Снимка на изгледа с примерни изходни данни

Няма изходни данни, тъй като подреденият масив не се извежда на екрана.

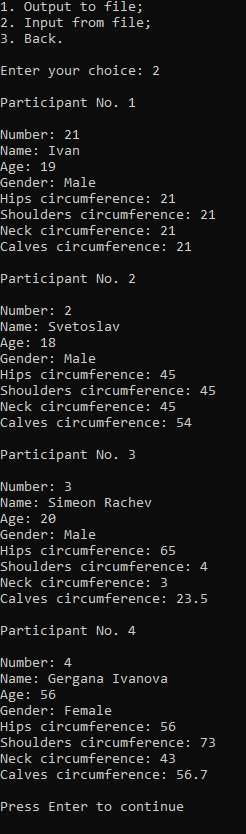
Условие F

Снимка на изгледа с примерни входни данни



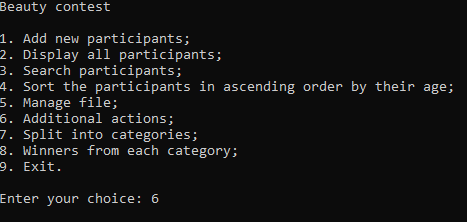


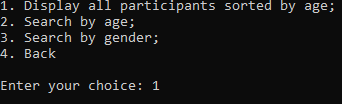
Снимка на изгледа с примерни изходни данни



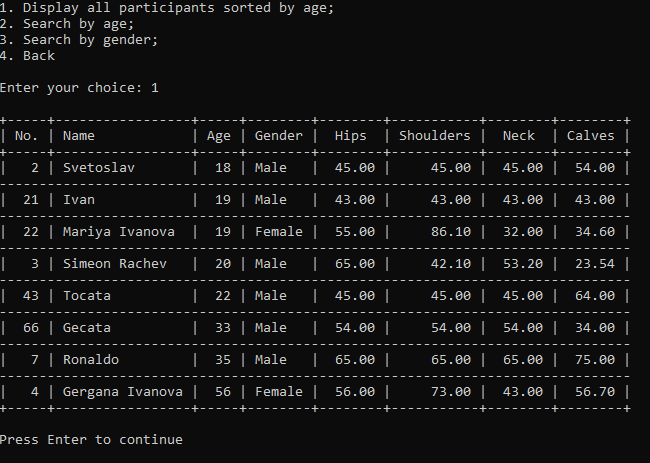
Допълнение първо

Снимка на изгледа с примерни входни данни

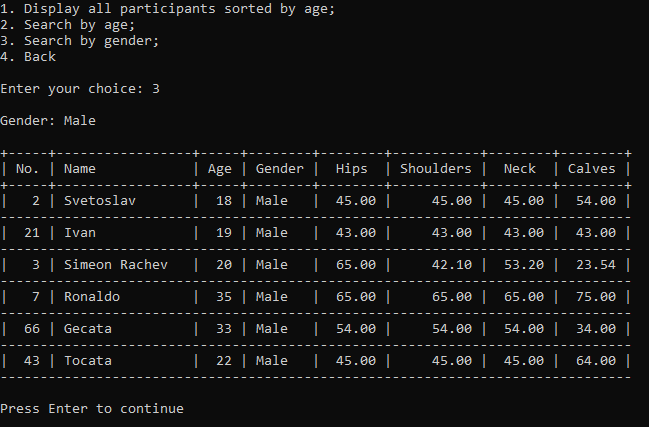




Снимка на изгледа с примерни изходни данни

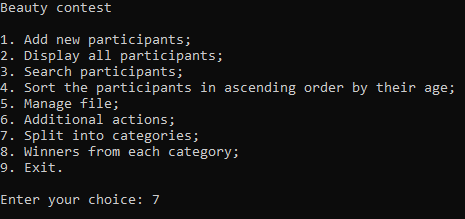


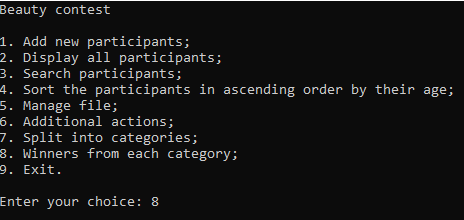




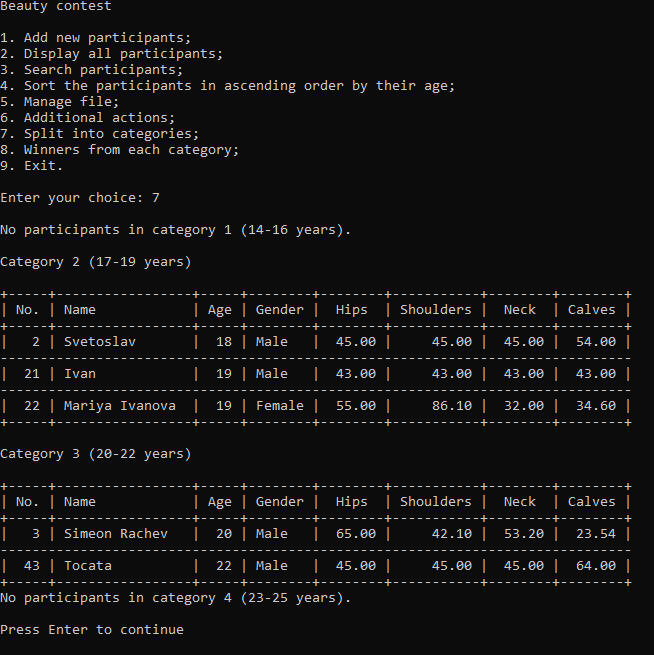
Допълнение второ

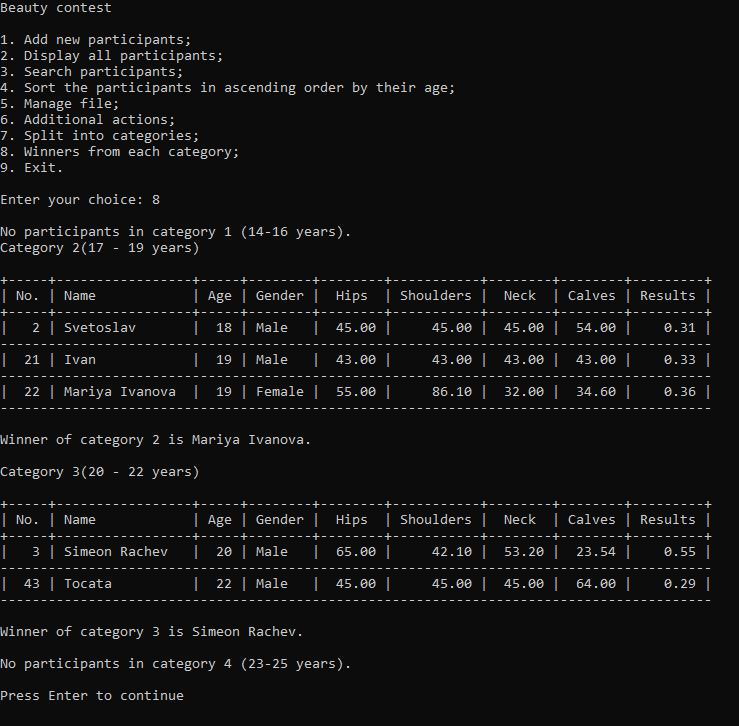
Снимка на изгледа с примерни входни данни





Снимка на изгледа с примерни изходни данни





Допълнение трето

Снимка на изгледа с примерни входни данни

Функциите за автоматично попълване на масива при стартиране на програмата и автоматично записване на данните във файл след затваряне нямат входни и изходни данни