TALLINN UNIVERSITY OF TECHNOLOGY

SCHOOL OF INFORMATION TECHNOLOGIES

Faculty of Computer Systems

223662MVEB

IAX0583 Programmeerimine I

**Files and Structss**

Homework no.1

Supervisor: Associate Professor Vladimir Viies

Tallinn 2022

**Copyright declaration**

**I have prepared this work independently. All the work of other authors used in the preparation of the work, important points of view, data from literary sources, and the seller are cited.**

**Rostyslav Boichuk 223662MVEB**

**Signature:**

**Contents**

**Statement of the task ...........................................................................4**

**Algorithm .............................................................................................5**

**Program code .......................................................................................7**

**Explanation of the program.................................................................9**

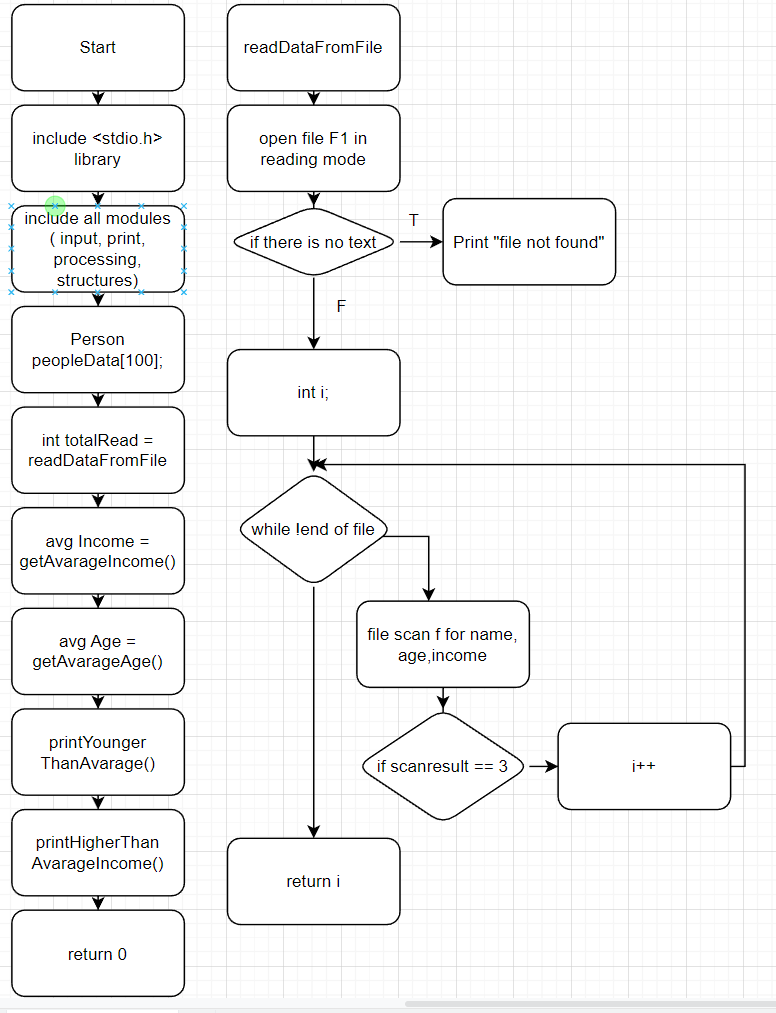
**Setting up the task**

V-5

The main goal of the task was to write the algorithm and code for its execution. The program should follow next requirements:

1. Data is read from a plain text file „F1.txt“ and stored as a structure. The data file must contain the given attributes:
   * Name - string
   * Age - integer
   * Income - floating point
2. Program will output:
   * To file „F2.txt“ all of the entries whose age is below the average.
   * To file „F3.txt“ all of the entries whose income is above the average

# **Algorithm**

****

**Diagram

Description automatically generatedA picture containing text, indoor

Description automatically generated**

**Program code  
Main:**

#include <stdio.h>

#include "input.h"

#include "print.h"

#include "structures.h"

#include "processing.h"

int main(void) {

Person peopleData[100];

int totalRead = readDataFromFile(peopleData);

float averageIncome = getAvarageIncome(peopleData, totalRead);

int averageAge = getAvarageAge(peopleData, totalRead);

printYoungerThanAverageAge(peopleData, totalRead, averageAge);

printHigherThanAverageIncome(peopleData, totalRead, averageIncome);

return 0;

}

**Input:**

#include "print.h"

int readDataFromFile(Person peopleData[]) {

FILE \*inputFile = fopen("F1.txt", "r");

if (inputFile == NULL) {

printf("File %s not found.\n", "F1.txt");

}

char lineBuffer[BUFFERSIZE];

int i = 0;

while (!feof(inputFile)) {

fgets(lineBuffer, sizeof(lineBuffer), inputFile);

int scanResult = sscanf(lineBuffer, "%s %d %f", peopleData[i].name, &peopleData[i].age, &peopleData[i].income);

if (scanResult == EOF) {

printf("End of file!\n");

} else {

}

if (scanResult == 3) i++;

}

return i;

}

**Processing:**

#include "processing.h"

#include "processing.h"

float getAvarageIncome(Person peopleData[], int totalRead) {

float totalIncome = 0;

for (int i=0; i<totalRead; i++) {

totalIncome += peopleData[i].income;

}

return totalIncome / totalRead;

}

int getAvarageAge(Person peopleData[], int totalRead) {

int totalAge = 0;

for (int i=0; i<totalRead; i++) {

totalAge += peopleData[i].age;

}

return totalAge / totalRead;

}

**Print(output):**

#include "print.h"

void printYoungerThanAverageAge(Person peopleData[], int totalRead, float avarageage) {

FILE \*outputFileAge = fopen("F2.txt", "w");

fprintf (outputFileAge, "File with people younger than average age: \n");

int i;

int totalAge = 0;

for (i=0; i<totalRead; i++) {

totalAge += peopleData[i].age;

}

float averageAge = totalAge / totalRead;

for (i=0; i<totalRead; i++) {

if (peopleData[i].age < averageAge) {

fprintf(outputFileAge, "%s %d %.2f \n", peopleData[i].name, peopleData[i].age, peopleData[i].income);

}

}

}

void printHigherThanAverageIncome(Person peopleData[], int totalRead, float avarageIncome) {

FILE \*outputFileIncome = fopen("F3.txt", "w");

fprintf (outputFileIncome, "File with people with higher income than average: \n");

int i;

for (i=0; i<totalRead; i++) {

if (peopleData[i].income > avarageIncome){

fprintf(outputFileIncome, "%s %d %.2f \n", peopleData[i].name, peopleData[i].age, peopleData[i].income);

}

}

}

**Stuctires file(to delare struct):**

#ifndef STRUCTURES\_H

#define STRUCTURES\_H

#define BUFFERSIZE 64

typedef struct Person {

char name[BUFFERSIZE];

int age;

float income;

} Person;

#endif

**Program explanation:**

In main file, we include all other files, namely their headers (.h files) as it eliminates the labor of finding and changing all the copies as well as the risk that a failure to find one copy will result in inconsistencies within a program. Then we call all the functions with appropriate arguments.

All Declarations of the files are in .h files what allows us to accses only to the function itself to avoid con flicts.

We use the input function(readDataFromFile) necessary data from file and store it in Struct „Person“.

In the processing file, we basicly declare functions to get avg int for age and avg float for the income of person.

In the print function we just compare each result stored in „person“ struct i order to meet the task requirements and write them down row by row to f2 and f3 files.