**Veldin Salcinovic**

**190302158**

**Project 1**

**USER MANUAL**

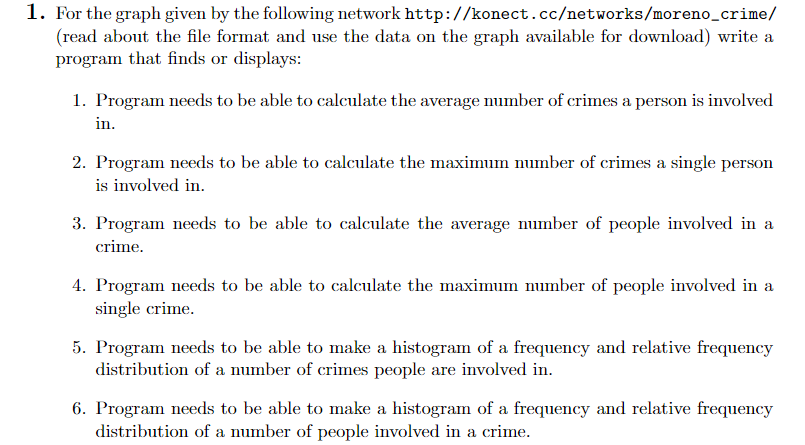
**FOR**

**“DISCRETE MATHEMATICS 2” PROJECT**

Part 1, Part 2, Part 3 and Part 4 of Project are done in: **C++ language.**

Part 5 and Part 6 of Project are done in: **R Studio.**

The problem for project 1 was:



**Table of Contents:**

**Page 1 ………………………………………… Initial Page**

**Page 2 ………………………………………… Table of Contents**

**Page 3 ………………………………………… Introduction and Overall View**

**Page 4 ………………………………………… Opening Files And Reading Data From Them**

**Page 5 ………………………………………… file(“sex”),(“crime”),(“role”)**

**Page 6 ………………………………………… PART1**

**Page 7 ………………………………………… PART2**

**Page 8 ………………………………………… PART3**

**Page 9 ………………………………………… PART4**

**Page 10 ………………………………………. PART5**

**Page 11 ………………………………………. PART5 in R**

**Page 12 ………………………………………. PART6 and PART6 in R**

**Page 13 ……………………………………….. SUMMARY**

**INTRODUCTION**

**The project I selected (project 1), I choose to do in C++ and RStudio. The first 4 parts of the project were calculations and readings from a given files and that is why I choose them to be done in C++. Last 2 parts of project are plotting graphs and I choose to do them in RStudio since it is much easier to plot them there than in C++ IDE.**

**On the next pages I will firstly explain how those things work in C++ and how the code is implemented. After that I will explain all parts form 1 to 6 respectively with explanations and pictures to understand it better.**

**Before I start, there are few things that might affect the code (in a bad way).**

* **For C++ part, I wrote code in CLion IDE, which has weird file opening process. Files that I used to read data from needs to be in their Projects folder and because of that I will try to navigate you through this User Manual and show you how I did it.**
* **Other thing is, again, related to files. This time it is about writing to a file, which I did for part 5 and part 6, where I wrote data from program to txt file so I could examine that data in RStudio.**

**FILES INSIDE .ZIP FOLDER EXPLAINED:**

Files that I worked on in my Project Folders. I am sending these files because in my C++ code it will open them with reference the names that are in folder.

**Text

Description automatically generated**

RStudio screenshots of graphs and R files that will automatically run the R Script

C++ code for first 4 parts of Project

User Manual

**OPENING FILES AND READING DATA FROM THEM**

**First thing is that I changed the names of files that I downloaded from web-page. I changed them to “names”, “sex”, “crime” and “role”. I did that because it is easier to organize in the code when choosing file path.**

**Text

Description automatically generatedA picture containing graphical user interface

Description automatically generated**

**From here, because of CLion, I had to move those files to CLion Project folder from where the information from the files could be read and stored into arrays.**

**After that is being done, the actual coding part begins.**

**For each file, I created different arrays which will store the data from the files respectively without changes in order.**

**I did this separately for every file and code for every one of them is pretty much the same.**

**Code below is for file “names” from which I stored all names into string array which will hold this data so it can be examined later on in the code**

**Text

Description automatically generated**

**There are also code for other files:**

**All of them are commented and I think there is nothing else to say about the code.**

**For file “sex”:**

**Text

Description automatically generated**

**For file “crime”:**

**Text

Description automatically generated**

**For file “role”:**

**Text

Description automatically generated**

**Next part of the code are actual solutions to the questions from Project 1.**

**I will go through all of them and explain them respectively:**

**Text

Description automatically generatedPART1:**

**This is the actual code which will print out solution to the user.**

**In this code, everything is commented, but still I will explain the logic behind the code.**

**Firstly, I declared array which will store data that we will get in for loops. The for loops are used to compare the data from different files. So for this part, we needed to compare the data from file “names” and file “crime”. From names, I didn’t actually selected names, but I selected their index reference, so for example if we want to check the first person it will check its index number in array, not actual name. So first loop will circle through persons from person 1 to person 829 and compare appearances of them in file “crime”, firs column. For every person separately, it will increment the avg value that I previously declared and later on add all of them. At the end it will divide them by 829, which is total number of people and we will get the approximate number of crimes a person is involved in.**

**The output result looks like this:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**PART 2**

**Text

Description automatically generated**

**For this part, I needed to calculate the number of crimes a selected person is involved in.**

**So to do so, I firstly created a code that will ask user to pick a number of some person, so for example if user chooses number 1, it will show data for person on position 1 in the file “names”.**

**The code is simple and similar to the previous one. It will again go through 2 for loops and compare the data from file “names” with data from file “crime” and increment the value of avg. Then, I added one if loop which will actually find only the criminals a person that user selected was involved in and print out the message about the person name and total number of criminals a person was involved in.**

**Code is correct and after entering multiple names, I calculated it in the file by hand and got the same result. It was a long procedure, but it was worth it.**

**Text

Description automatically generated**

**As you can see, user will be asked to enter persons number and then the information will be printed.**

**PART 3**

**Text

Description automatically generated**

**For part 3, I needed to calculate the average number of people in a crimes.**

**So here I firstly needed to calculate the total number of crimes that we have in file “crimes”, which are located in second column.**

**To do so I created small for loop which circled through all of them in second column and found the largest one. The largest number was 551, which means that there are 551 criminal cases in this project. After I did that, I commented this part so it doesn’t confuse user when running the program. If you want to check it yourself, you can just remove the comment marks on begging and on the end of the desired code.**

**For the rest, it is similar as in part 1. It will circle through all of crimes and check the number of their appearances and then calculate the average appearance of people in crimes.**

**The result should look like:**

**A picture containing graphical user interface

Description automatically generated**

**PART 4**

**Text

Description automatically generated**

**In this part, I needed to calculate the number of people involved in crime which user selects. So if user selects crime number 5, I needed to output the number of people that were involved in that crime.**

**The code is again similar as code in part 2 because it is the same story just opposite sides.**

**It will circle through for loops and compare all of the values and after for loops, in if loop, it will check only the information and number of people for crime case that was selected by user.**

**So, user will be prompted message to select crime number and after user writes crime number, the message about that crime will be printed.**

**In the program it will look like:**

**Text

Description automatically generated**

**This means that crime number 35 has 5 people involved in it.**

**Before I go to part 5,**

**The program is made so when you run it, it will do all the tasks at the same time, so in one run, the user will be asked to enter all those number and every result will be printed respectively from part 1 to part 4.**

**For part 5 and part 6, the code in C++ was only to take data from the code or from some files and write it in new file that I created and that will be attached with the code in .zip folder. Those file, named “part5” and “part6” will be used, opened and examined in RStudio, but data that will be stored in them will be written and stored by C++ code (PART 5 and PART6)**

**PART 5**

**For part 5, I needed to plot histogram of frequency and relative frequency distribution for number of crimes a person is involved in. So basically, this is part 1, just plotted in R Studio.**

**To complete this part, firstly I created one file called part5 in CLion Projects folder. I will use that file to store data that I compute from my C++ code. After doing that I calculated the number of people that were involved in one crime, two crimes and so on. I did that by comparing the values and storing them into variables c1….c25.**

**After gathering all that information, I exported and wrote that information into file “part5” by the following code:**

**The file is initially empty, but after you run the code, data will be automatically written.**

**Text

Description automatically generated**

**Next thing is to open RStudio and insert the following code into it, but before you do that, you should put this file “part5” into your working directory:**

**part5 <- read.table("C:/Users/Veldin/CLionProjects/DiscreteMath/cmake-build-debug/part5", quote="\"", comment.char="")**

**Number\_of\_crimes\_Frequency <- part5$V2**

**hist(Number\_of\_crimes\_Frequency, main="Histogram for number of crimes people are involved in", xlab="Crimes", ylab = "Number of people", xlim=c(0,900),las=1,**

**breaks=5)**

**Number\_of\_crimes\_person\_is\_involved\_in\_Relative\_Frequency <- round(100\*part5$V2/sum(part5$V2), 2)**

**hist(Number\_of\_crimes\_person\_is\_involved\_in\_Relative\_Frequency, main="Histogram for number of crimes people are involved in", xlab="Crimes", ylab = "Number of people", xlim=c(0,100),las=1,**

**breaks=5)**

**I attached RStudio files as well to .zip folder.**

**Graphical user interface, text, application

Description automatically generatedThose are screenshots from RStudio:**

**PART 6**

**For part 6, I needed to plot histogram of frequency and relative frequency for number of people involved in a crime.**

**To do so I created for loop which will extract all data from file “crime” from 2nd column into one file called “part6”.**

**The code in C++ looks like:**

**Text

Description automatically generated**

**All crimes will be stored in one file called “part6” and then that file will be examined in RStudio. Same process and everything.**

**Code from RStudio:**

**part6 <- read.table("C:/Users/Veldin/CLionProjects/DiscreteMath/cmake-build-debug/part6", quote="\"", comment.char="")**

**part6$part6\_Cat<-cut(part6$V1,**

**breaks=seq(0.5, 551.5, 1))**

**Number\_of\_people\_involved\_in\_crime\_Frequency <- table(part6$part6\_Cat)**

**hist(Number\_of\_people\_involved\_in\_crime\_Frequency,main="Histogram for number of people involved in a crime", xlab="Number of people", ylab = "Crimes")**

**round(100\*table(part6$part6\_Cat)/sum(table(part6$part6\_Cat)), 2)**

**hist(Number\_of\_people\_involved\_in\_crime\_Frequency,main="Histogram for number of people involved in a crime", xlab="Number of people", ylab = "Crimes")**

**Files for PART 6 and PART 5 will be uploaded as well in .zip folder.**

**SUMMARY**

**The files that will be uploaded inside .zip folder:**

* **Names.txt**
* **Sex.txt**
* **Crime.txt**
* **Role.txt**
* **Part5.txt (empty)**
* **Part6.txt (empty)**
* **Main.cpp**
* **Part 5 project. R**
* **Part 6 project. R**
* **User Manual**
* **Histogram Pictures (4 pictures)**

**All of this is done with one run in C++ IDE and hopefully it will all work. I something doesn’t work or if something doesn’t open, probably the reason is difference in IDE or some typo mistake.**

**In the C++ code I added some testing to check if the results are correct for some parts and they were working properly. You can either delete those tests, skip them or even try them.**