TUNNITÖÖ „Programmeerimine II“

Nimi Kuupäev

Koostada üldine algoritm (kus on näha programmimoodulite andmevahetus) ja sellele vastav programm, keeles C mille abil(6p):

1. Tekstifailist C1 sisestatakse kirjed struktuuriga:

* Nimi – string,
* Synna – kirje struktuuriga:
* P, K ja A – täisarvulised (vastavalt sünnipäev, –kuu- ja –aasta);

2. Klaviatuurilt sisestatakse nädalapäeva nimi(või järjenumber nädalas) X;

3. Kuvatakse kõik aastad ***intervallist Synna.A…Synna.A+100*,** millal sünnipäev on X nädalapäeval, arvestades et 1.jaanuar 1900 oli esmaspäev.

ALAMÜLESANDED+:(a’1p)

1.Juhul kui failis vigased kirjed, need väljastada, kuid tööd jätkata;

2.Juhul kui sünnipäev on juubel, tähistada see eraldi “!!!”;

3.Juhul nimi ei koosne ees- ja perenimest, väljastamisel lugeda sisestatu perenimeks ja lisada eesnimeks Mari või Jyri.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fail C1

andrus varik 02,03,1950

Vvvvvvvvv01,01,2000

georg suur 47,01,2001

jaan ilus 05,05,1999

peep haak 12,12,1988

mia viiding 13,14,2003

mart juur 07,09,1930

ilme oks 29,02,1997

 SUBTASKS+:(a'1p)

1. In case of incorrect entries in the file, issue them, but continue the work;

2. If the birthday is a jubilee, celebrate it separately "!!!";

3. If the name does not consist of a first and last name, when issuing, read what is entered as the last name and add Mari or Jyri as the first name.

Work 1 *NB! each task is a separate function*

Mark was interested when their teachers have birthdays. He looked through their CVs and written down their birthdates to text file ( C1.txt ). The files consists of multiple rows each one representing one person. In every row there are name surname and birthdate (Separated by spaces; Day, month and year - by comma): **Andrus Varik 02,03,1950**

Task 0[2point]**Create a general algorithm (where the data exchange of program modules can be seen) and a program corresponding to it, in C language, to solve the following tasks**

Task 1 [1 point]:

The program must asks user filename with data. If the file does not exist, the program has to ask again until the file exists( but not more than 3 times). For example:

 ***Enter filename: C8.txt: File does not exist*** ,***try again you can do it 2 more times***

Task 2 [1 point]

Read the records from the file and print the number of results into console. During reading the file, the program has to validate data and fix it, at the same time printing an error message for every invalid row found, indicating row number and updated data.. Program has to check:

***\*If name or surname is missing insert Juri or Mari randomly instead of it***

***\*If such day - month combination does not exist – print and delete from file (For example 32 of January, or 29 of February 2001 does not exist)***

Task 3 [2 points]

Find weekday of the birthdate for every person and print them into console. Either as a number (1,2,..7) or as a string ("Monday", "Tuesday", ...) **Hint: 1 January 1900 was Monday**

Mark has graduated from univerisy, but sometimes comes back to visit his teachers. However, he can come only on weekday X (1,2,..,7). Find all teachers who have birthday on weekday X from year Y to year Z. Names and dates into console in chronological order.

If it's jubilee (age is divisible by 10), add [age] to the end of the line. For example:

 ***Enter weekday X: 4 Enter year Y: 2000 Enter year Z: 2030***

*Possible result:*

*2,3,2000 - Andrus Varik [50]*

*23,9,2004 - Mart Juur*

*2,3,2006 - Andrus Varik*

*23,9,2010 - Mart Juur [30]*

*2,3,2017 - Andrus Varik*

*23,9,2021 - Mart Juur*

*2,3,2023 - Andrus Varik*

*23,9,2027 - Mart Juur*

*2,3,2028 - Andrus Varik*

Constrains

name and surname are between 1 and 20 characters. Name may be missing

day , month are any 2 digit integers

year is an integer between 1900 and 2100

[day],[month],[year] is separated by comma and always present

Number of records in the file is between 1 and 50

Requirements

The logic of the program has to be split into functions. Prototypes are obligatory.

For example, reading the file, validating the data,printing the results, etc.

Names of the functions has to describe what they do.

Try to format the output in a way that it is easy to read read and understand. Use your design skills :)