UNIVERSITY OF PARMA

Department of Engineering and Architecture
Degree course in Computer, Electronic and Communications Engineering
PRACTICAL TEST IN COMPUTER SCIENCE & PROGRAMMING LABORATORY
1st Nivôse CCXXXIV

Name: _____ Surname:_____ Matr:_____ Workstation: _____

Write a program using the C language (name the project with your student <**ID**>) that behaves as described below. The available time is 120 minutes. At the end of the time, the saved files on **U:** are going to be automatically collected. Additional documents, files… are available in **T:\Bertozzi**, it is recommended to use WordPad to read text files.

The reform in the field of weights and measures, imposed by the French Revolution through the decimal metric system, was at the time also extended to the calendar. In fact, the Gregorian calendar was replaced by the Revolutionary Calendar (**RC**). This consisted of 12 months of 30 days each. Each year also had 5 or 6 additional days, for a total of 365 or 366 days, depending on whether the year was leap or not. The names of all the months were changed, and the use of the new dates was enforced starting from 22/09/1792, which corresponds to the "new" 01/01/01.

Starting from the provided code skeleton, implement a program that converts the Gregorian dates contained in the ASCII file "dati.txt" and saves them into a second file according to the following procedure:

1. (10) Open and read line by line the file "dati.txt", which contains on each line a date and the description of an event in the format:
   `<dd>/<mm>/<yyyy>;<event description (max 200 characters)>`
2. For each line read, using the data obtained:
   a. (5) Invoke a function `checkdata()` (**to be implemented**) that returns *false* if the given date is earlier than 22 September 1792. If it returns *true*, proceed with the conversion as described below; otherwise, continue reading from the file.
   b. (1) Invoke the already defined function `giorni()` which returns the number of days elapsed since 22/09/1792 (start of the RC).
   c. (8) Use the number returned in the previous step to calculate how many years have passed since the start of the RC, for example by counting how many times it is possible to subtract 365 (or 366 every 4 years).
   The result of this procedure will be the RC year, as well as the number of days elapsed since the beginning of that year.
   d. (3) Based on the days elapsed since the beginning of the year, calculate the month and the day of the month, or determine whether the date falls within one of the 5 or 6 complementary days of the RC.
   e. (3) Save the newly converted date into the file "rivcal.txt" together with what was read from the file in step #1, using the format:
   `<day> <month name> of the year <year>: <event description>`
   or
   `<name of complementary day> of <year>: <event description>`
3. Modify point 2.c so as to consider that "leap" years are those divisible by 4 but not by 128.

The code must be developed in the indicated order (optionally, point #3 may be developed together with point #2.c). For each implemented step, verify correct functionality. The grading stops at the first point that is not correctly implemented.

The provided code skeleton, in addition to the previously described `giorni()` function, contains constant string arrays with the names of the months and days to be used for the solution and, for those who wish to use it, a function to convert the year into Roman numerals.