Problem 1 (20 points)

In this problem you will help a (benevolent) social planner to understand how employment histories are distributed in the simulated economy. Data for this problem is provided in hw1p3.csv. hw1p3.csv contains following variables:

- date_of_birth - given in the format "YYYYMM", 196106 means that individual is born in March, 1961;
- gender - categorical variable indicating gender with usual labels;
- exp_by_1996 - work experience in years until 1996;
- d199601 - d201412 - dummy variable showing whether an individual worked in a given month or not (0 - does not work, 1 works).

Questions:

1. Calculate number of missing values for each variable in the dataset:

- Using loops;
- Using apply (functions such as apply, tapply, lapply, etc.) family;
- Using data.table.

2. Calculate mean work experience before 1996 by year of birth and gender:

- Using loops;
- Using apply (functions such as apply, tapply, lapply, etc.) family;
- Using data.table.

3. Plot mean work experience before 1996 as a function of year of birth both for women and men in the same plot. Your plot should have clearly labelled x axis and y axis title and legend so that we can understand, which function is for men and which is for women.
4. Count the number of missing values in variables d199601 - d201412 by individual:

- Using loops;
- Using apply (functions such as apply, tapply, lapply, etc.) family;
- Using data.table.

5. Calculate total work experience starting 199601 in months. If an individual has a missing value in any of the variables d199601 - d201412 then total work experience starting 199601 for that individual should be equal to NA.
6. Write a function, which calculates work experience starting 199601 for a given individual by a given date. That is, your function should take as input all variables d199601 - d201412 for a given individual, date in the format "YYYYMM" and return how many months a given individual worked by this date. If NA are encountered in the variables d199601 - date provided by you, function should return NA.
7. Write a function, which calculates total work experience reached at a given age. That is, your function should take as arguments exp_by_1996, d199601 - d201412, date_of_birth and age and calculate how many months of work experience individual has by a particular age. If individual has not reached this age or reached this age before 199601 your function should return "FIELD_NOT_FOUND".
8. Calculate for all individuals total work experience in months reached by 201112.
9. Calculate for all individuals total work experience in months reached by age 60.

Problem 2 (30 points)

In this problem you will help social planner to figure out how many children a given women has at a particular date.

- Social planner does not know how many children a given women has at a particular date but she knows whether she paid childcare benefits to a given woman at a particular date.
- She knows that if in a given month there is 1 in the data, then in this month woman received childcare benefits;
- On the other hand, if in a given month there is 0 in the data then in that month woman did not receive childcare benefits;
- She also remembers that sometimes in the data there are mistakes - if the difference between two subsequent sequences of 1s is less than 5 (that is there are less than 5 0s in between two subsequent sequences of 1s), then a woman received childcare benefits for the same child and it is not the case that a new child was born

Your task is to help social planner to understand how many children a women has at any given point in time To do that you need to write a function, which implements what a social planner knows:

1. As inputs your function should take a vector of 0s and 1s;
2. Every time you see a sequence of 1s in the data you need to increase the number of children by 1;
3. Be careful with the two subsequent sequences of 1s, where the difference between them is less than 5 (i.e. when there are less than 5 0s in between them, then it is the same child and not a new child);
4. To help you social planner provides some examples of what your function should return:
    - Input: c(0,0,0,1,1,1,0,0)
    - Output: 0 0 0 1 1 1 1 1
    - Input: c(1,1,1,1,0,0,0,0)
    - Output: 1 1 1 1 1 1 1 1
    - Input: c(0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1)
    - Output: 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2
    - Input: c(0,0,0,0,1,1,1,1,0,0,1,1,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,1)
    - Output: 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
    - Input: c(0,1,0,0,0,0,1,1,0,0)
    - Output: 0 1 1 1 1 1 1 1 1 1
5. Functions, which might be helpful:
    - rle();
    - diff();
    - cumsum();
    - which().

Problem 3 (30 points)

- In the folder hw1p5 you will find data on public procurement for one of Russian regions. More specifically you will find data on notifications;

Questions:

1. Write a function, which converts a given archive in the notifications folder into a list of data.tables. Each data.table should contain following fields:

- notificationNumber;
- versionNumber;
- createDate;
- placingWay.code;
- placingWay.name;
- order.placer.regNum;
- lots.lot.products.product.code;
- lots.lot.customerRequirements.customerRequirement.maxPrice;
- Name of the child.

   If there are several fields with the same name please join them together using "&&&&".

2. Apply this function over all archives in the notifications folder. Note that notifications folder also contains a daily subfolder. You should consider zip files located in the daily subfolder too.
3. Convert this list of lists into (by doing necessary flattening before) into one big data.table. Keep only those observations, where the name of the child equals notificationZK or notificationEA.
4. Keep unique observations by notificationNumber, versionNumber and createDate.
5. Calculate the following:
    - Sum of lots.lot.customerRequirements.customerRequirement.maxPrice in 2011, 2012 and 2013;
    - Number of auction procedures in 2011, 2012 and 2013;
    - notificationNumber where the buyer bought the most goods (as proxied by the number of lots.lot.products.product.code);
    - Distribution of placingWay.code by year.