C projects

Global variables may not be used in any of the assignments.

Write user friendly programs:

- When user is requested to input a value, the program must clearly state what the user is expected to enter
- Error message shall be displayed if the user enters an invalid value.

Reports must print data in formatted columns where printed data is aligned by columns. For example, when printing numbers make the field width large enough to hold largest number that you are going to print.

Take care that array bounds are not violated - especially with strings!

Do not use “magic numbers” - use #define for array sizes and any other limits/constants that you may need.

Start by thinking what type of input you need to ask from the user. Write functions that perform error checking for types that you need.

Use functions to structure your program.

In the following assignments “database” means a data structure (array, linked list, etc.) in the memory that contains the data.

1. Password keeper

Store passwords by site/address, search by address, save passwords to file with simple “encryption” based on master key that is asked from the user. The master key may not be stored anywhere. If the user forgets the master key the passwords can’t be recovered. Passwords are decrypted only when displayed and plaintext password must be wiped from memory after display.

Print a report of sites and passwords.

Program encrypts the passwords by performing xor-operation between master key and the password.

For example master key is "metropolia" and password is "hetz68#extraSecretP@sswrd".

Encryption is performed by xoring each character in the password with a character from the master key. Master key is repeated consecutively over the password. The encryption may produce zero as a result of encryption so you need to store the length of the password to be able decrypt the password later.

Principe of operation:

Xor these two characters. Result is the encrypted character. Then advance to next pair etc.

```
metropoliametropoliametropoliametropoliam
hetz68#extraSecretP@sswrd
```

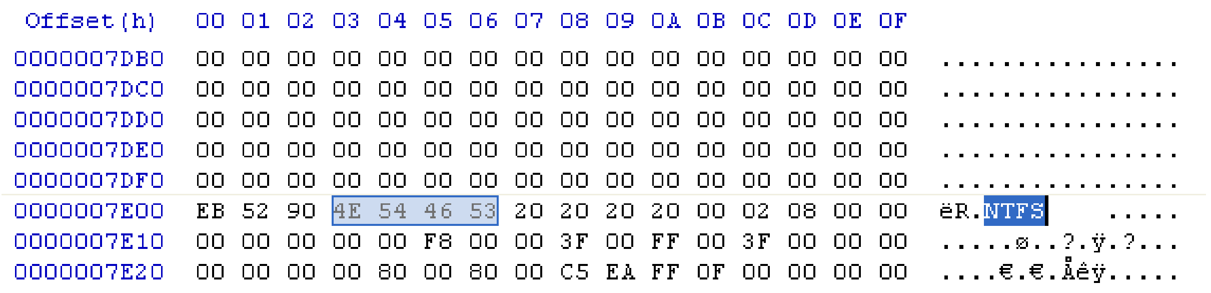
Running the “encryption” twice using the same master key should returns password to its original state.

2. Hex editor/patch tool

View and edit files in hexadecimal mode. Saves log of changes in another file. Apply previous log to the current file.

The editor must have at least the following commands:

- Display content of the file. User gives offset from beginning of the file and program displays couple of lines of data (for example 4 rows, 16 bytes on each row) and displays data both in hexadecimal and text if possible (unprintable characters will print a dot). See example below. Note that you don’t need to have any graphics printing plain text is enough.



```
 Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

0000007DB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007DF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000007E00  EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëR.NTFS     .....
0000007E10  00 00 00 00 00 F8 00 00 3F 00 FF 00 3F 00 00 00  .....ø..?.ÿ.?...
0000007E20  00 00 00 00 80 00 80 00 C5 EA FF 0F 00 00 00 00  ....€.€.Åêÿ.....
```

- Command to change file contents (offset + new value(s))
- Command to save changes
- Command to apply previous change log to this file

To create a change log, save only commands that modify file content (command to make changes, save command)

3. Stock tracker

Write a program that keeps record of stock market trades. Program stores following information about each trade:

Description	Type and/or size
Name of the company	max. 80 characters
Number of shares	int
Purchase price of the shares	float/double
Date of purchase	string (or whatever type you think is appropriate)
Sell price of the shares	float/double
Date of sale	string (or whatever type you think is appropriate)

The program must have (at least) following operations:

1. initialize (empty the database)
2. save database to file
3. read database from file

4. buy shares
5. sell shares
6. print a report which prints list of all shares that you still own followed by the total price of the shares.
7. Print a report of all sold shares. Display profit/loss per sale. Print total profit/losses at the end of the report.

4. Auto loan calculator

Tool for car salesman: enter value of old car (one that is traded in), price of the new car, length of the loan. Print a report that shows size of monthly payment, etc.

https://koppa.jyu.fi/avoimet/maths/matyl/peruskurssi/luentomateriaalia/tasaeralaina.pdf

https://www.vertex42.com/ExcelArticles/amortization-calculation.html

Save each loan offer into a file and allow them to be loaded from file for editing and/or printing later.

https://financeformulas.net/Loan_Balloon_Balance.html

5. File protector/viewer

Encrypts files with (simple) encryption algorithm. See assignment 1 for encryption algorithm

1. Encrypt a file:
    a. Ask user to enter filename
    b. Ask user to enter password
    c. Encrypt file
    d. Store filename in database
2. View file:
    a. Ask user to enter filename
    b. Load file to memory
    c. Check database to see if file is encrypted
    d. Ask password and decrypt data if the file is encrypted
    e. View the data
3. Remove encryption
    a. Ask filename
    b. Verify that file is listed as encrypted
    c. Decrypt and save the file
    d. Remove filename from database

Keeps a database of encrypted files. When user views a file check log to see if password is needed. Ask password and decrypt before viewing or view if file is not encrypted.

6. Calendar

The program must have (at least) following operations:

1. initialize (empty the database)
2. save database to file
3. read database from file
4. add an event: user enters description, date, time and duration of an event

5. remove an event
6. print a report which prints all events sorted by date and time
7. search for a free time in the range of dates that user specifies

7. File compression

Implement RLE-encoding/decoding of file. Write log of compression results (name, original size, CRC, compressed size, name of compressed file).

The encoding is based on encoding the multiple occurrences of the same byte with an encoding that saves number of repeating bytes instead of individual bytes. The encoding works as follows:

0x80 is a special character that indicates encoded sequence. It is immediately followed by the byte to repeat and number of repetitions. For example, a sequence of seven 'T's would be encoded as: 0x80 0x07 0x54. From this we can see that encoding sequences of less than three same characters increases the result file size instead of decreasing size.

Occurrence of 0x80 is always encoded regardless of the length. There is one exception in the encoding, when number of repetitions is set to zero it indicates a 0x80. A single 0x80 would be encoded as 0x80 0x00, two 0x80s would be encoded as 0x80 0x02 0x80 and three 0x80s as 0x80 0x03 0x80 and so on.

The encoded file must contain a small header that carries the CRC of the original file so that decompressed data can verified.

Usage of program:

`rle pack filename` (compresses the specified file)

`rle unpack filename` (decompresses the specified file)