

# C projects for smart systems

The program must work on an LPCXpresso.

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

Program must use structures and save data to disk in binary format.

Write user friendly programs:

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

## 1. Black Jack

<https://en.wikipedia.org/wiki/Blackjack>

Must save game log to a file and print a report from the log.

## 2. Battleship

[https://en.wikipedia.org/wiki/Battleship\\_\(game\)](https://en.wikipedia.org/wiki/Battleship_(game))

Must save game log to a file and print a report from the log.

## 3. 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). 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@sswrđ".

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.

Principle of operation:

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

  
m  
h

metropoliаметрополиаметрополиаметрополиам  
hetz68#extraSecretP@sswrđ

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

## 4. Hex editor/patch tools

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

```
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  òR.NTFS .....
0000007E10  00 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 00  ....€..Ïéÿ.....
```

- 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)

## 5. Expense tracker

Write a program that keeps record of money you have spent. Program keeps track of your financial situation by storing both your incomes and expenses. Program stores following information about each financial transaction:

Description	Type and/or size
Type of transaction (income/expense)	integer

Description of transaction (for example: "this month's salary", "1 kg rice, bag of curry, 400 g chicken"	max. 80 characters
Amount of money	float
date of transaction	string (or what ever type you think is appropriate)

The program must have (at least) following commands:

1. initialize (clear all records)
2. save financial records to file
3. read financial records from file
4. add money
5. spend money
6. print a report which prints all transactions and prints separate sum of incomes and expenses and difference between sums (positive difference means you are saving, negative meaning that you are spending more than you earn)

## 6. Memory view / edit tool

View and edit RAM on LPCXpresso. Similar to HEX editor except that edits take place in RAM. Saves log of changes in a file.

RAM starts at address 0x2000000 and the size of RAM is 0x8000 (32 kB). You can get a pointer to memory the following way:

```
uint8_t *memory = (uint8_t *) 0x2000000;
```

Then you can access memory like an array of bytes. Start your tests by viewing memory without making any changes. There is no memory protection so you can easily mess up your own stack and/or heap. Start making changes in the middle of the available memory. It is most likely to be free and does not make your program crash immediately.

## 7. 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 printed again later.

[https://financeformulas.net/Loan\\_Balloon\\_Balance.html](https://financeformulas.net/Loan_Balloon_Balance.html)

## 8. File protector/viewer

Encrypts files with (simple) encryption algorithm. Same password for all files. Keeps a log 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.

## 9. Calendar

Save date/time of events. Produce listings. Search for a free time on given date(s).

## 10. 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 compressing it.

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 be verified.