Exercise 12. *Deadline: 8 June 2009.*

COURSE OF C++ PROGRAMMING LANGUAGE

XOR cipher

University of Wrocław
Institute of Computer Science *Paweł Rzechonek*

Exercise

Write a program, which will encrypt/decrypt data stored in a file using a secret password. The name of the input file and of the output file and the secret key pass into your programm via command line arguments.

Implement relatively simple encryption algorithm called *XOR cipher*. The algorithm encrypts data using bitwise XOR on a data stream and a key. Observe that applying the same key on the encrypted data decrypts the original data. Define a funftion `XORcipher()`, which will encrypt/decrypt data; this function should throw the exception `ios_base::failure` automatically if any problem with a stream appears (invoke the method `exceptions()` at the begining (set your own state) and at the end (return the initial state) of the function).

```
void XORcipher (string key, istream &in, ostream &out) throw (ios_base::failure);
```

Test your program by encrypt a short text file and after then decrypt the result using the same passord. Compare the second result with the original file.

Example

For the key $K = \langle ABCD \rangle$ and the text $T = \langle how\ are\ you \rangle$, the correct result is $\langle$*)-4d 0&d8-6*$\rangle$.

the key	A	B	C	D	A	B	C	D	A	B	C
the text	h	o	w	␣	a	r	e	␣	y	o	u
the result	)	-	4	d	␣	0	&	d	8	-	6

For each $i = 0 \ldots |T| - 1$ perform bitwise operation *XOR* on a pair of characters $T_i \oplus K_{i \bmod |K|}$.

	$\langle A \rangle$ 01000001	$\langle B \rangle$ 01000010	$\langle C \rangle$ 01000011	$\langle D \rangle$ 01000100	...
$\oplus$	$\langle h \rangle$ 01101000	$\langle o \rangle$ 01101111	$\langle w \rangle$ 01110111	$\langle ␣ \rangle$ 00100000	...
	$\langle) \rangle$ 00101001	$\langle - \rangle$ 00101101	$\langle 4 \rangle$ 00110100	$\langle d \rangle$ 01100100	...

Hint

Pass some information into the program (into the `main()` function) via command line arguments. There are two special built–in arguments, `argv` and `argc`, that are used to receive command line arguments. The `argc` parameter holds the number of arguments on the command line and it is an integer. It is always at least 1 because the name of the program qualifies as the first argument. The `argc` parameter is a pointer to an array of character pointers. Each element in this array points to a command line argument.

```
int main (int argc, char *argv[])
{
    // ...
}
```

Suggestion

Partition your code into the header and source files.

Hint

Some information about XOR cipher can be found on the webpage:

`http://en.wikipedia.org/wiki/XOR_cipher`