Unit 9: Pointers and Arrays

Pointers are one of C's most important features. In this unit we consider the use of pointer as pointer variables, pointers as arguments to functions, pointers as return values, and pointers to arrays or elements of arrays, and using array names as pointers. Some more applications of pointers will follow in later units. Relevant for this unit are Book Chapters 11 and 12. In chapter 12 pointer arithmetic is described in detail, wich is important for programming the second variant of the programming project.

Project u9: Variants of Palindrome

Write a program that reads a message, then checks whether it's a palindrome (the letters in the message are the same from left to right as from right to left). Ignore all characters that are not letters only relevant are letter A-Z,a-z. A single character is a palindrome, and the empty message is considered a palindrome as well (this includes a message that consists of only non-letter characters).

```
% palindrome1
% Enter a message: He lived as a devil, eh?
% Palindrome
% palindrome2
% Enter a message: He lived as a devil, eh?
% Palindrome
% palindrome1
% Enter a message: Anne, I stay a day at Sienna.
% Palindrome
% palindrome1
% Enter a message: Madam, I am Adam.
% Not a palindrome
%
```

The maximum length of the message is 40 characters. Reading of the message should stop when more than 40 characters are read, or when the character `'\n'` is read (additional characters are ignored). The two possible messages printed by the program are "Palindrome" and "Not a palindrome".
Hand in two versions of this program (both behave exactly the same, but are implemented differently):

- Programming Project u9 Version 1: `u9_palindrome1.c`: Use integer variables to keep track of positions in the array and use the array access operator (brackets '[' and ']') for accessing elements of the array.
- Programming Project u9 Version 2: `u9_palindrome2.c`: Use pointers instead of integers to keep track of positions in the array.

Hint: Have a look at the example programs reverse.c and reverse3.c; the first one showing how to print a reversed string by using only integers to keep track of positions, the latter

only using pointers to keep track of positions. These programs use the same operations that are needed in the two different versions of the palindrome check. For checking whether a character sequence is a palindrome, you need to access the array from both sides and, while ignoring non-letter characters, compare the letters.
Refer to the provided test-cases and compare the output of your program with the provided output of the test-cases. Note that both versions are represented by the test-cases. Your program should behave exactly like the test-cases before you hand it in. Hand in the programs palindrome1.c and palindrome2.c.