



## Computer Science 3B

### Practical Assignment 07

Assignment date:

2024-09-19

Deadline:

2024-09-19 17h05

Marks: 100

---

This practical assignment must be uploaded to [eve.uj.ac.za](http://eve.uj.ac.za) before 2024-09-19 17h05.

Late<sup>1</sup> or incorrect submissions will not be accepted and will therefore not be marked. You are not allowed to collaborate with any other student. Plagiarism is not tolerated. All submissions are tested for plagiarism.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present.

The reminder page includes details for submission. Please ensure that ALL submissions follow the guidelines. The reminder page can be found on the last page of this assignment.

---

Happy with the company's R&D progress thus far, **BeatWatch** decided to digress and look at improving its string functionality management. They decided to start with a basic prototype for some string manipulations. For example, they would like to be able to replace certain characters in a string and be able to reverse strings.

Write an **80x86** assembly program with the following:

#### Main program structure (requirements):

1. Use *registers* that seems relevant for the task (for e.g., **ecx**, **esi**, **al**, **etc.**)
2. Create *global variables* only. Besides the below, any supporting variables you can create at your discretion.
3. For the character replacement, create two variables **searchChar** and **replaceChar**. The **searchChar** variable will hold the character to be searched for in the user provided string (see *point 4*), and the **replaceChar** variable will hold the character to replace the search character. *For now, you can hardcode these values (see the examples provided in the table below).*
4. For the string reversal, the user must be *prompted* to provide a string as input (for example a sentence, or paragraph) and the entered string can be stored in an **inputStr** variable and the reversed string in a **reversedStr** variable and a message *outputted* to the screen (see the examples provided in the table below).
5. You must create **3** functions:
  - a. **main** proc: this is the entry and exit into the program
  - b. **replaceCharacters** proc: called from the main proc, this is to perform the basic character replacement.

<sup>1</sup> Alternate arrangements for exceptional circumstances will be posted on eve.

- c. **reverseString** proc: called from the main proc, this is to perform the string reversal.

**Testing set** – Use these values to test your program

| Prompt          | Input String   | search Char | replace Char  | outputMsg   |
|-----------------|--|-------------|---------------|---|
| Enter a string: | simple example\$   | #           | !             | <b>Reversed string:<br/>#elpmaxe elpmis</b>   |
| Enter a string: | New%moon   | %           | (blank space) | <b>Reversed string:<br/>noom weN</b>  |
| Enter a string: | !egaugnal<br>gnimmargorp msam<br>gnisu gnidoc dna<br>gninrael evol I<br>\$revewoh<br>\$noitalupinam gnirts<br>htiw dnuora gniyalp<br>yojne I | \$          | ,             | <b>Reversed string: I<br/>enjoy playing<br/>around with string<br/>manipulation,<br/>however, I love<br/>learning and coding<br/>using masm<br/>programming<br/>language!</b> |

### Example output

```
Enter a string: !egaugnal gnimmargorp msam gnisu gnidoc dna gninrael evol I $revewoh $noitalupinam gnirts htiw dnuora gniyalp yojne I
Reversed string: I enjoy playing around with string manipulation, however, I love learning and coding using masm programming language!
```

### Mark sheet

|    |  |                    |
|----|--|--------------------|
| 1. | Design   | <b>[10]</b>        |
| 2. | Prompt Implementation  | <b>[05]</b>        |
| 3. | Character Search and Replace Implementation                              | <b>[15]</b>        |
| 4. | String reversal Implementation   | <b>[20]</b>        |
| 5. | Message output Implementation  | <b>[05]</b>        |
| 6. | 3 Procs Implementation   | <b>[15]</b>        |
| 7. | Structure and layout (no temporary/local variables, relevant data types) | <b>[05]</b>        |
| 8. | Commenting   | <b>[05]</b>        |
| 9. | Correct execution  | <b>[20]</b>        |
|    |  | <b>Total [100]</b> |

## NB

Submissions that **do not assemble** will be capped at 40%!

Practical marks are awarded subject to the student's ability to explain the concepts and decisions made in preparing the practical assignment solution.

(Inability to explain code → inability to be given marks.)

Execution marks are awarded for a correctly functioning application and not for related code.

## Reminder

Your submission must follow the naming convention below:

SURNAME\_INITIALS\_STUDENTNUMBER\_SUBJECTCODE\_YEAR\_PRACTICALNUMBER

**Example:** Berners-Lee\_TJ\_209912345\_CSC03B3\_2024\_P05

|                       |             |                         |         |
|-----------------------|-------------|-------------------------|---------|
| <b>Surname</b>        | Berners-Lee | <b>Module Code</b>      | CSC03B3 |
| <b>Initials</b>       | TJ          | <b>Current Year</b>     | 2024    |
| <b>Student number</b> | 209912345   | <b>Practical number</b> | P05     |

Your submission must be **a single zip (compressed) file!**

Your submission must include the following:

| File   | Naming                | Folder | Purpose   |
|--------|-----------------------|--------|---|
| Design | STUDENTNUMBER_P05.pdf | docs   | Contains your program design. All files must be in <b>PDF</b> format. Your details must be included at the top of any <b>PDF</b> files submitted <sup>0</sup> . |
| Source | STUDENTNUMBER_P05.asm | src    | Contains all relevant source code. Your details must be included at the top of the source code <sup>0</sup> .   |

## Multiple uploads

Note that only **one** submission is marked. If you already have submitted once and want to upload

a newer version then submit a newer file with the same name as the uploaded file in order to overwrite it.

<sup>0</sup>Failure to correctly indicate your details will result in a penalty.