UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS132 - Imperative Programming
Practical 4

1 Introduction

Deadline: 9th May, 20:00

1.1 Objectives and Outcomes

The objective of this practical is to practice reading from files, arrays, characters and the Switch-Case statement.

1.2 Submission

You will have **10 submissions** where the total for the practical is **5 marks**. Submit your code to Fitchfork before the deadline. Students are **strongly advised** to submit well before the deadline as **no late submissions will be accepted**.

1.3 Plagiarism

Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to the COS132 Study guide. If you have questions regarding this, please ask one of the lecturers, to avoid any misunderstanding.

2 Practical Requirements

2.1 Turing Machine

You are required to write a program that will implement a Turing Machine. To simplify the program the Turing Machine will have a finite tape length, the maximum length possibly will be 10000 characters but that does not imply each tape will be of exactly that length.

You will need to read in a string from a provided file named **"turing.txt"** into an array. Once the string has been read into the array you will need to traverse the array using the rules provided until the accepted or rejected state is reached. Once the Turing Machine has either rejected or accepted the tape you will need to write to a file named **"tape.txt"**

the final state of the tape. It is important to note the spelling of the file names and the file types, everything is lower case and a normal text file. If you get this wrong it may cause you to lose marks.

2.1.1 Tape Symbols

The tape of the Turing Machine consists of the following characters:

```
U X P
```

Where "U" is the empty character. Just because the character exists it does not mean it will be in every tape.

2.1.2 Tape Motions

The Turing Machine has four different motions. It can either move left, move right, move to the accepted state or move to the rejected state.

When the tape is moving right or left this means you have to increment or decrement the index of the array to go to the next or previous element. To move right is to increment and move left is decrement.

Any unwritten transitions will go to the reject state. If the the motion moves to an empty element it is populated with the empty character "U".

If the tape is accepted the following message is outputted:

```
The tape was accepted
```

If the tape is rejected the following message is outputted:

```
The tape was rejected
```

2.1.3 Operations

When the internal state (this is the current element at the index we are looking at in the array) of the turing machine is the following symbols the given operation needs to be applied:

```
q0
    If value is P change to U and move Right to q1
q1
    If value is P change to X and move Right to q2
    If value is X move Right to q1
    If value is U move Right to accept
q2
    If value is X move Right to q2
    If value is P move Right to q3
    If value is U move Left to q4
```

```
q3
    If value is P change to X and move Right to q2
q4
    If value is P move Left to q4
    If value is X move Left to q4
    If value is U move Right to q1
```

The following diagram is given to help represent the above described operations:

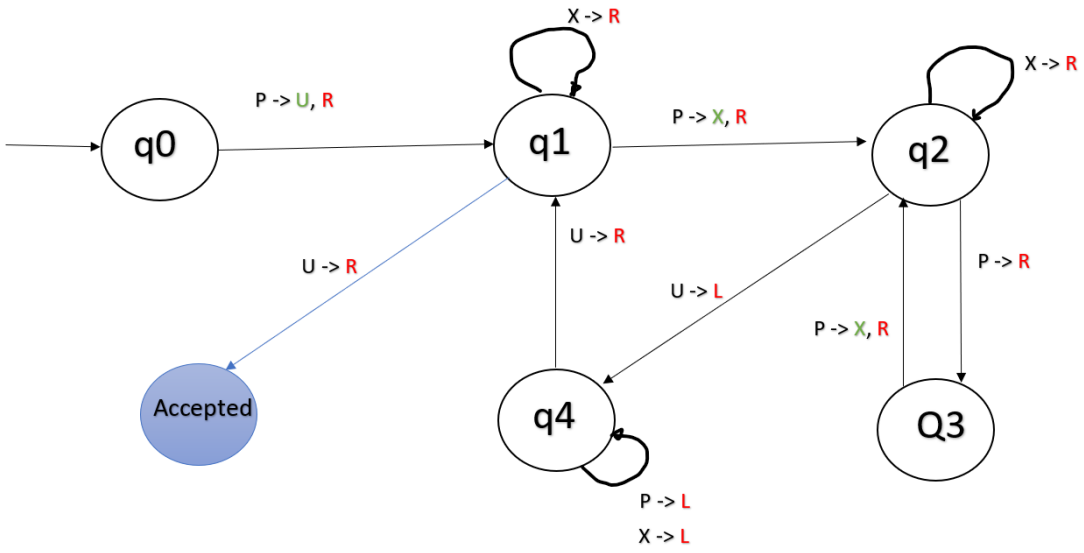

Figure 1: Turing Operations

The green values is what the current values will change to (if none it does not change) and the read is the move operation where R is right and L is left.

2.1.4 Example

The given file contents:

```
PPPP
```

The tape is going to look as such:

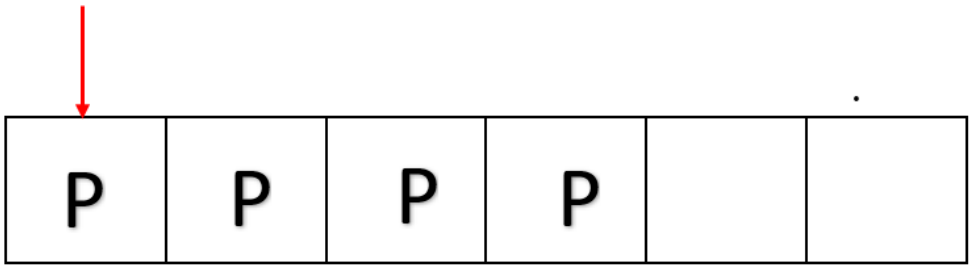

Figure 2: Turing Tape

The array of characters that represents the tape will contain the following:

```
tape[0] -> P
tape[1] -> P
tape[2] -> P
tape[3] -> P
```

For your futher knowledge this is how some of the tape was changed after running over the steps:

```
PPPP
UPPP
UXPP
UXPP
UXPXU (This step is repeated multiple times with going to multiple states)
UXXXU (This step is repeated multiple times with going to multiple states)
```

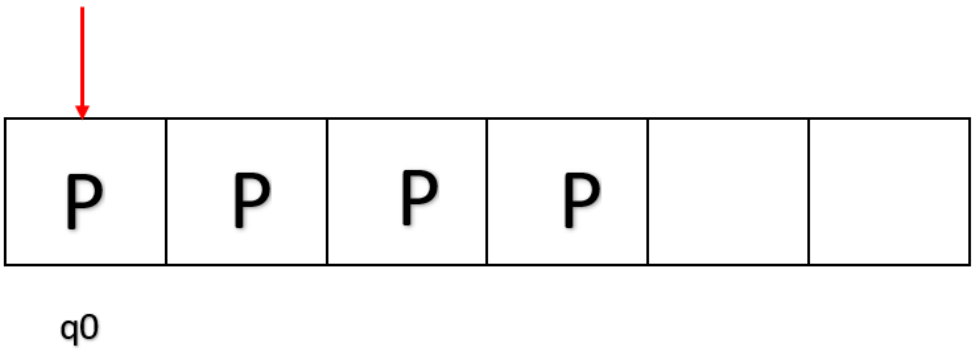

Figure 3: Tape Starts at q0 and changes P to U

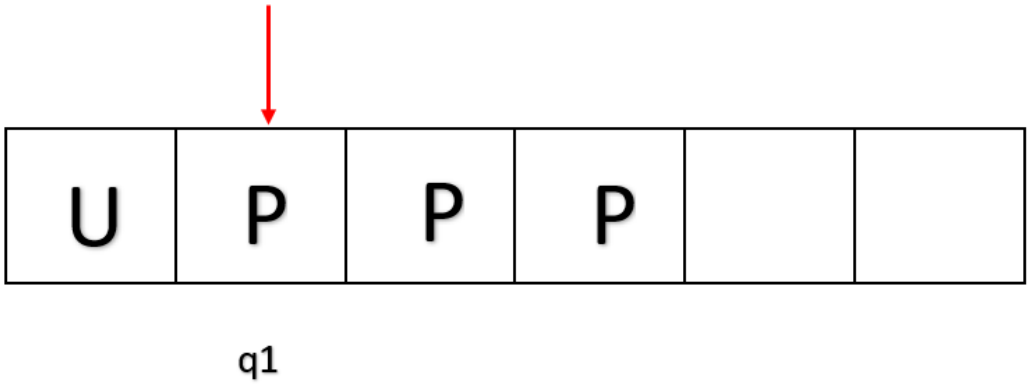

Figure 4: The tape is now q2 and does not change the value

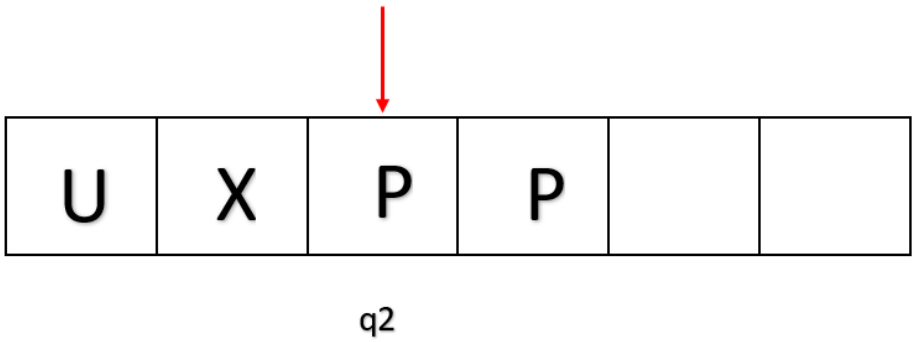

Figure 5: Turing Tape

Example of the required program:

```
The file contained: PPPP
The final tape was: UXXXU
The tape was accepted
```

To help aid you in implementing you may use the following links:

- Wikipedia: Turing machine
- Turing Machine Simulator
- Turing Machine Example

It is importatnt to realize ONE SPECIFIC Turing Machine, which is characterized precisely by the Transition-Table in the specification document given to them. In other words, the students are NOT creating a general TM-Emulator for "any" arbitrary TM. This specific TM is "hard-coded" in the SWITCH-statements in their C++ program.

2.2 Marks

Marks will be awarded as follows:

```
Reading and writing from Files: 2 marks
Correct implementation of Turing Machine: 2 marks
Correct accept/reject message: 1 mark
```

As with the previous practical you will not receive your marks until after the deadline.