String Matching Assignment

KPR Lakmal

21001065

Introduction

In this assignment I use horspool algorithm to implement the code.

In addition to that I have included four regular expression patterns and I have implemented four functions for each regular expression pattern.

Regular expression pattern I have used.

1. \`.\` (dot):

   - Pattern: \`b.t\`

   - Matches: "bat", "bet", "bit", "bot", "but", etc.

   - Explanation: The dot matches any single character except a newline.

2. \`?\` (question mark):

   - Pattern: \`colou?r\`

   - Matches: "color" and "colour."

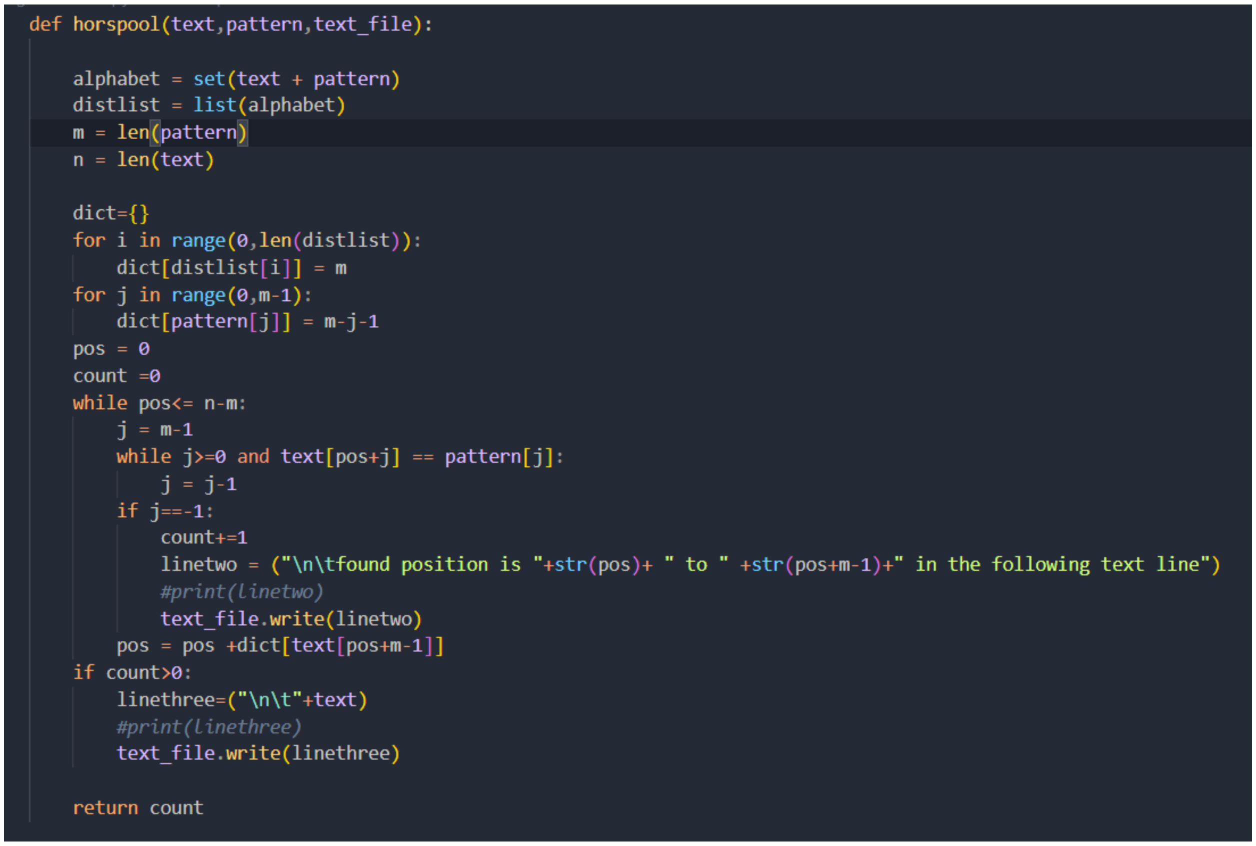
   - Explanation: The question mark matches zero or one occurrence of the preceding character, in this case, the letter "u."

3.\`^\` are used to specify the beginning of a line or string.

4.\`$\` are used to specify the end of a line or string

Why HorsPool.

Because Horspool algorithm matches from the end of the pattern and when there is a mismatch (baschatacter) found it's not going to search for the next right most occurrence of the bad character. Instead it's searching for the right most occurrence of the last character of the current text window in the pattern. There it skips lot of unwanted matches at once. In our case the text file contains a huge amount of texts. High number of skips decreases the time to search for a pattern in a text. Therefore Horspool algorithm is an efficient algorithm in this scenario.

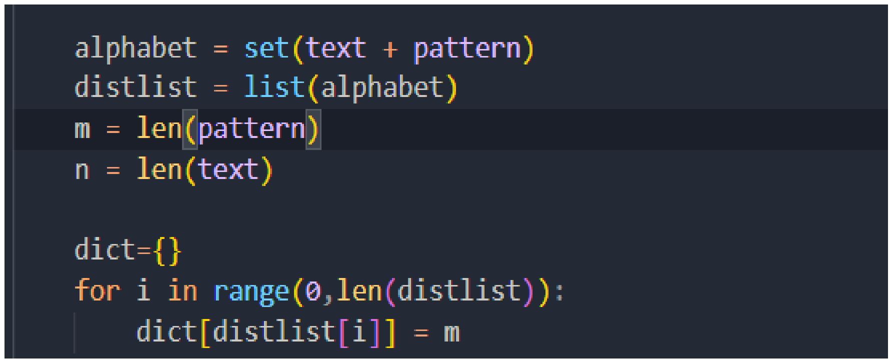


```python
def horspool(text,pattern,text_file):

    alphabet = set(text + pattern)
    distlist = list(alphabet)
    m = len(pattern)
    n = len(text)

    dict={}
    for i in range(0,len(distlist)):
        dict[distlist[i]] = m
    for j in range(0,m-1):
        dict[pattern[j]] = m-j-1
    pos = 0
    count =0
    while pos<= n-m:
        j = m-1
        while j>=0 and text[pos+j] == pattern[j]:
            j = j-1
        if j==-1:
            count+=1
            linetwo = ("\n\tfound position is "+str(pos)+ " to " +str(pos+m-1)+" in the following text line")
            #print(linetwo)
            text_file.write(linetwo)
        pos = pos +dict[text[pos+m-1]]
    if count>0:
        linethree=("\n\t"+text)
        #print(linethree)
        text_file.write(linethree)

    return count
```

Above code describes how the horspool algorithm works. I have created a function named horspool and took text , pattern and output file as inputs.

And if there exist any matching in text then write matching line and matching position in output file.

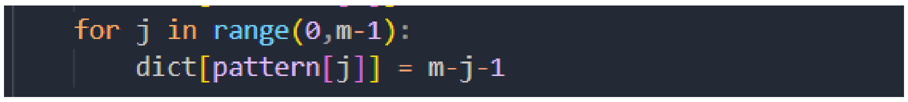
How it works:



```
alphabet = set(text + pattern)
distlist = list(alphabet)
m = len(pattern)
n = len(text)

dict={}
for i in range(0,len(distlist)):
    dict[distlist[i]] = m
```

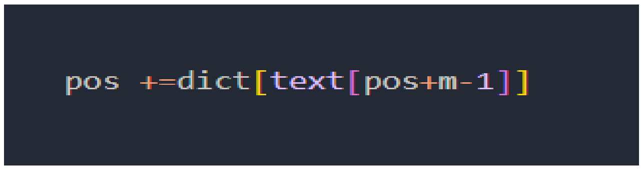
As we know the horspool algorithm uses the last character of the current text window instead of bad character. In my solution, I've used a dictionary in python to keep track of the right most position of each character in the pattern so that it will be our HpBc table. For the HpBc table I've considered characters in both text and the pattern. Because there might be some characters in the text which cannot be found in the pattern anywhere and vice versa. So I've created a set and then converted it to a list so that I can access each character by an index. And for each character we need to instantiate a key in the dictionary. Initially all the keys have the same value which is the length of the pattern.

After that we need to calculate real right most position of the each character.



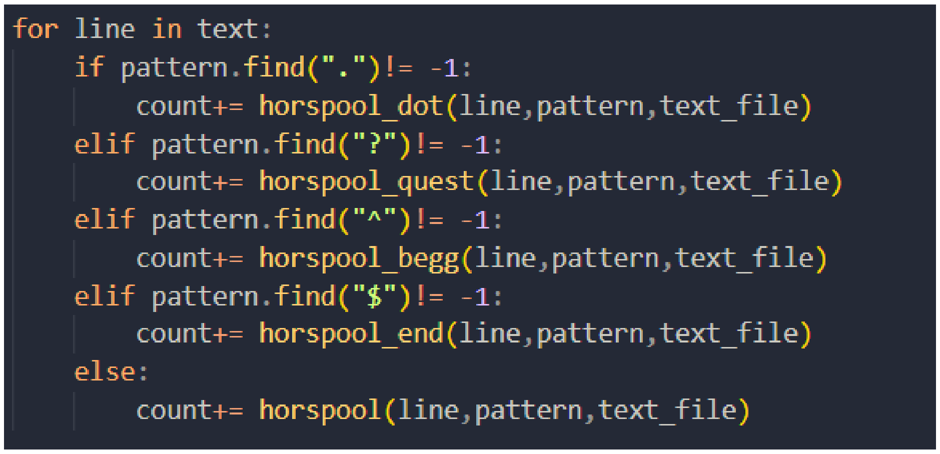
```
for j in range(0,m-1):
    dict[pattern[j]] = m-j-1
```

Then we need two index variables to iterate thrplough the text and the pattern. For the text I've used variable "pos" which is initially equals to 0 and on each iteration of pos, we define another variable j which is initially equals to lentht(pattern)-1. Starting from the last character of the current text window. Value of j will be decreased if the two characters are equals or else value of pos will increase if there was a mismatch. To index "pos" to next suitable point we're using the pre-calculated HpBc table.



```python
pos +=dict[text[pos+m-1]]
```

And the following part of the code check the pattern has any regular expression character. If there exist any regex character then its function call using if else statements.



```python
for line in text:
    if pattern.find(".")!= -1:
        count+= horspool_dot(line,pattern,text_file)
    elif pattern.find("?")!= -1:
        count+= horspool_quest(line,pattern,text_file)
    elif pattern.find("^")!= -1:
        count+= horspool_begg(line,pattern,text_file)
    elif pattern.find("$")!= -1:
        count+= horspool_end(line,pattern,text_file)
    else:
        count+= horspool(line,pattern,text_file)
```

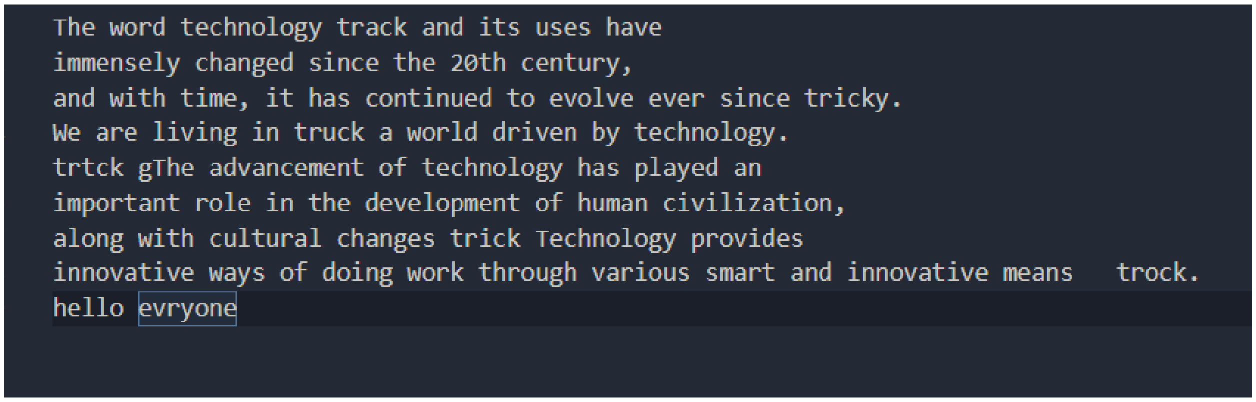
I have implemented four various functions for each regex character.

dot(.)	=> call horspool_dot()
question mark(?)	=> call horspool_quest()
dollor sign ($)	=> call horspool_begg()
^ mark (^)	=> call horspool_end()

These are some Test and results

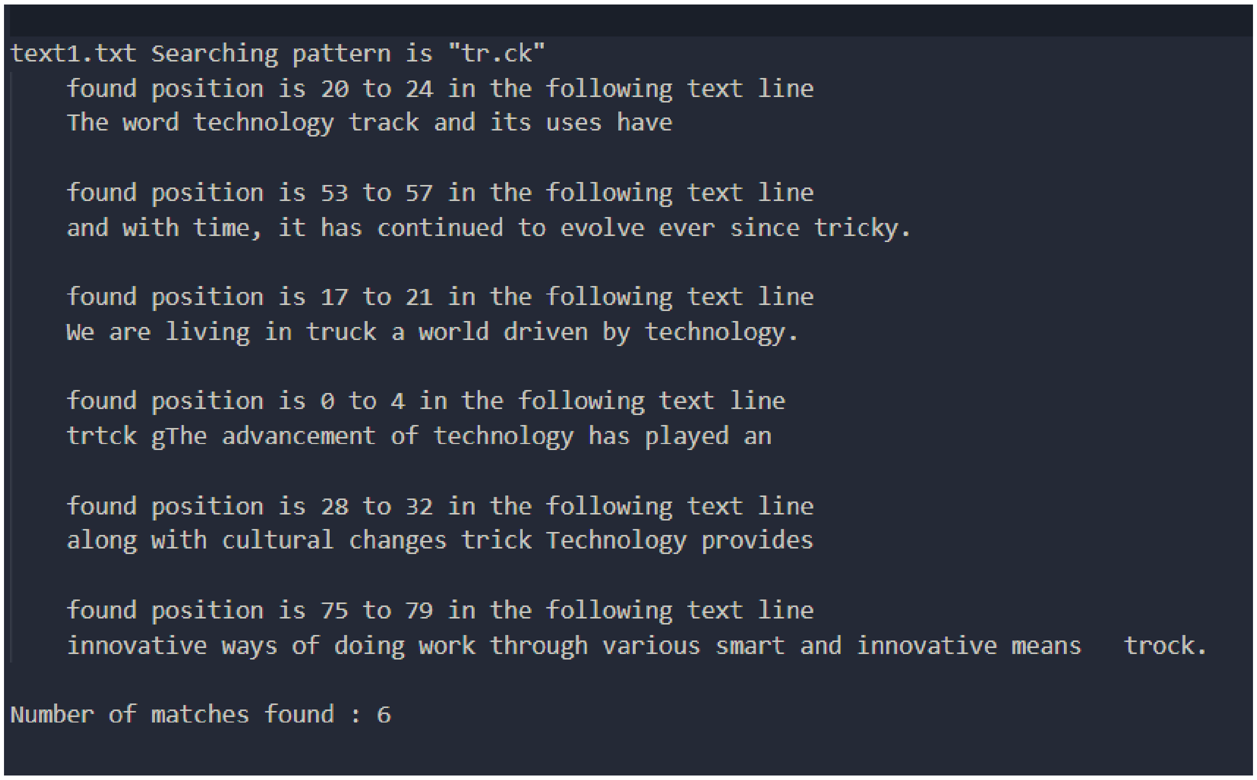
Test One

1. **Pattern.txt => " tr.ck "**
2. **Text.txt**



```
The word technology track and its uses have
immensely changed since the 20th century,
and with time, it has continued to evolve ever since tricky.
We are living in truck a world driven by technology.
trtck gThe advancement of technology has played an
important role in the development of human civilization,
along with cultural changes trick Technology provides
innovative ways of doing work through various smart and innovative means   trock.
hello evryone
```

3.Output1.txt

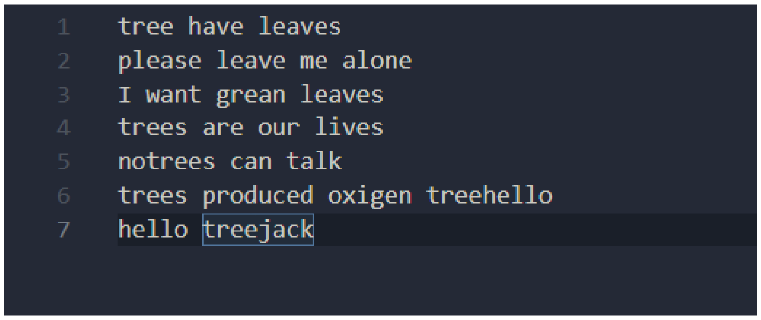


```
text1.txt Searching pattern is "tr.ck"
    found position is 20 to 24 in the following text line
    The word technology track and its uses have

    found position is 53 to 57 in the following text line
    and with time, it has continued to evolve ever since tricky.

    found position is 17 to 21 in the following text line
    We are living in truck a world driven by technology.

    found position is 0 to 4 in the following text line
    trtck gThe advancement of technology has played an

    found position is 28 to 32 in the following text line
    along with cultural changes trick Technology provides

    found position is 75 to 79 in the following text line
    innovative ways of doing work through various smart and innovative means   trock.

Number of matches found : 6
```

Test two

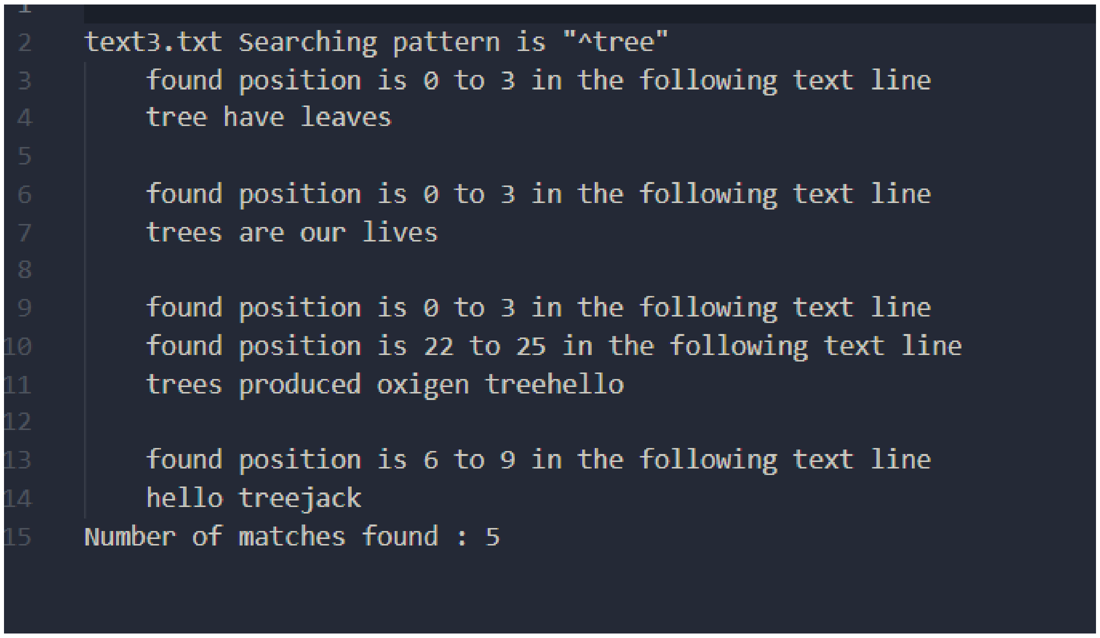
1.Pattern.txt => “ ^tree ”

2.text.txt



```
tree have leaves
please leave me alone
I want grean leaves
trees are our lives
notrees can talk
trees produced oxigen treehello
hello treejack
```

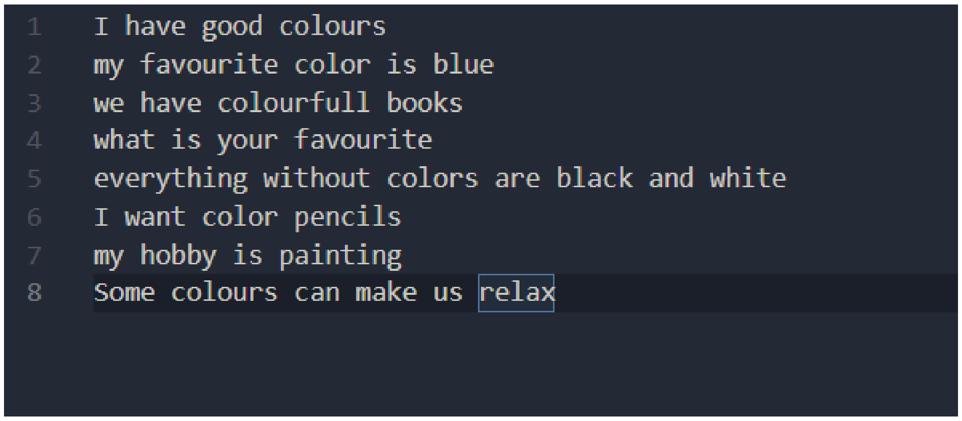
3.output.txt



```

text3.txt Searching pattern is "^tree"
    found position is 0 to 3 in the following text line
    tree have leaves

    found position is 0 to 3 in the following text line
    trees are our lives

    found position is 0 to 3 in the following text line
    found position is 22 to 25 in the following text line
    trees produced oxigen treehello

    found position is 6 to 9 in the following text line
    hello treejack
Number of matches found : 5
```

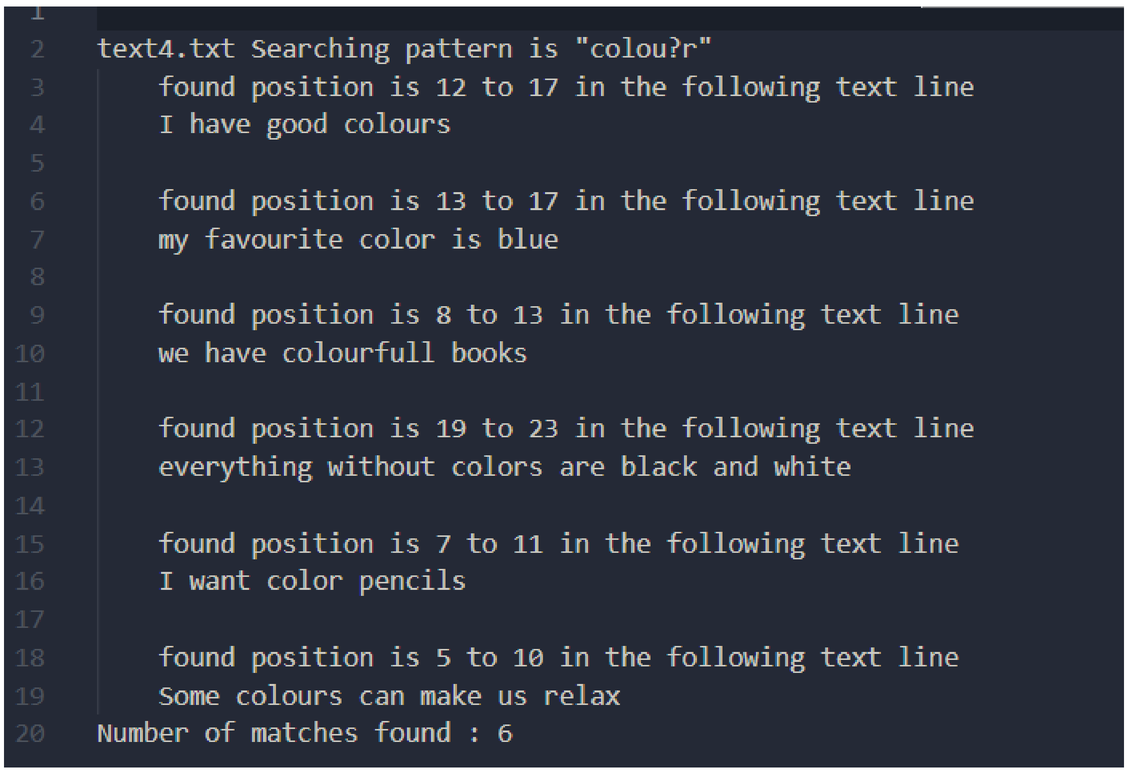
Test Three

1. **Pattern => “colou?r”**
2. **Text.txt**



```
I have good colours
my favourite color is blue
we have colourfull books
what is your favourite
everything without colors are black and white
I want color pencils
my hobby is painting
Some colours can make us relax
```

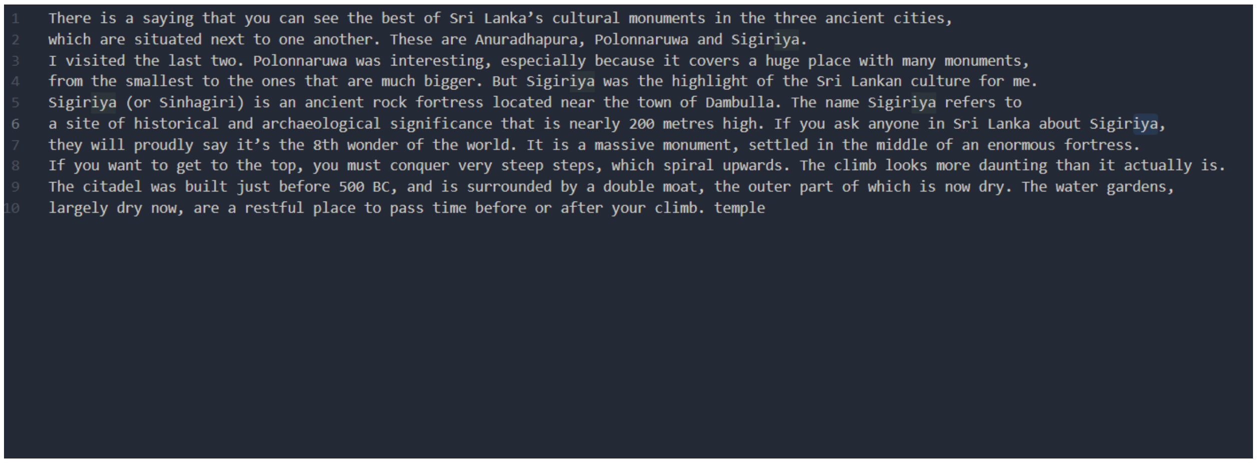
3.output.txt



```

text4.txt Searching pattern is "colou?r"
    found position is 12 to 17 in the following text line
    I have good colours

    found position is 13 to 17 in the following text line
    my favourite color is blue

    found position is 8 to 13 in the following text line
    we have colourfull books

    found position is 19 to 23 in the following text line
    everything without colors are black and white

    found position is 7 to 11 in the following text line
    I want color pencils

    found position is 5 to 10 in the following text line
    Some colours can make us relax
Number of matches found : 6
```

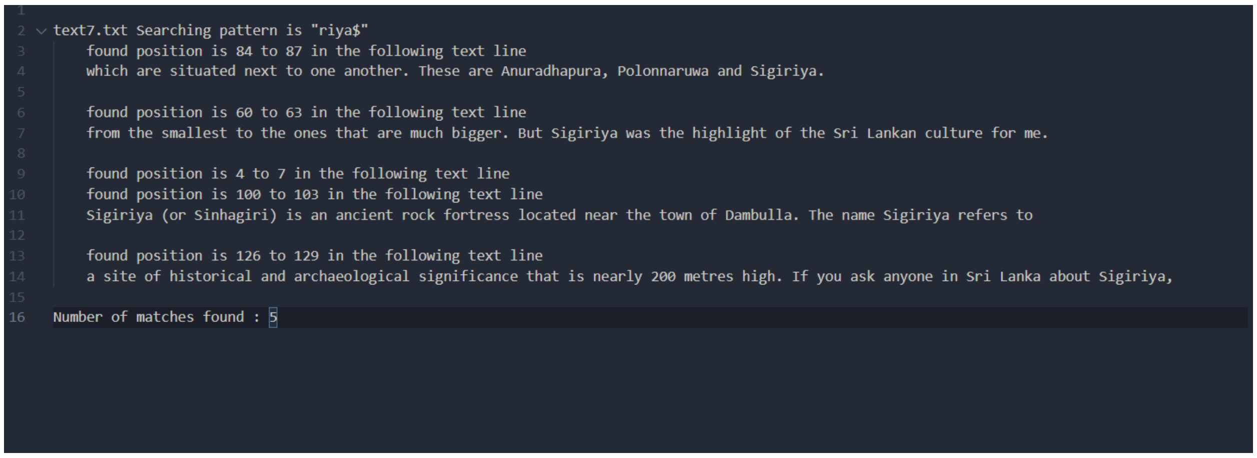
Test Four

1. **Pattern => “riya$”**
2. **Text.txt**



```
There is a saying that you can see the best of Sri Lanka’s cultural monuments in the three ancient cities,
which are situated next to one another. These are Anuradhapura, Polonnaruwa and Sigiriya.
I visited the last two. Polonnaruwa was interesting, especially because it covers a huge place with many monuments,
from the smallest to the ones that are much bigger. But Sigiriya was the highlight of the Sri Lankan culture for me.
Sigiriya (or Sinhagiri) is an ancient rock fortress located near the town of Dambulla. The name Sigiriya refers to
a site of historical and archaeological significance that is nearly 200 metres high. If you ask anyone in Sri Lanka about Sigiriya,
they will proudly say it’s the 8th wonder of the world. It is a massive monument, settled in the middle of an enormous fortress.
If you want to get to the top, you must conquer very steep steps, which spiral upwards. The climb looks more daunting than it actually is.
The citadel was built just before 500 BC, and is surrounded by a double moat, the outer part of which is now dry. The water gardens,
largely dry now, are a restful place to pass time before or after your climb. temple
```

3.output.txt



```

text7.txt Searching pattern is "riya$"
    found position is 84 to 87 in the following text line
    which are situated next to one another. These are Anuradhapura, Polonnaruwa and Sigiriya.

    found position is 60 to 63 in the following text line
    from the smallest to the ones that are much bigger. But Sigiriya was the highlight of the Sri Lankan culture for me.

    found position is 4 to 7 in the following text line
    found position is 100 to 103 in the following text line
    Sigiriya (or Sinhagiri) is an ancient rock fortress located near the town of Dambulla. The name Sigiriya refers to

    found position is 126 to 129 in the following text line
    a site of historical and archaeological significance that is nearly 200 metres high. If you ask anyone in Sri Lanka about Sigiriya,

Number of matches found : 5
```