**IT1140 PYTHON PROGRAMMING**

**B**

**CYCLE TEST 1**

**MAX MARKS: 25 DATE: 02.08.2016**

**PART – A (ANSWER ANY FIVE) 5\*4 = 20 marks**

1. How do you run the underlying operating system terminal commands (like “dir” in windows) from python.

import os

os.system(‘dir /B’)

(or) os.popen(‘dir /B’).read()

1. Brief the difference between process forks and threads.

Forks – Separate copy of the process. Independent process space

Threads – Parallel pieces of work inside the same process. Share the same process space.

1. List what pdb command will you use for each of the below listed operation.

|  |  |
| --- | --- |
| **Operation** | **Command** |
| To see the last successfully executed command | up |
| To execute the program till the next break point | continue |
| To inspect the content of a variable | !<variable\_name> |
| To execute line by line without entering into the functions. | step |

1. List and brief the different phases in Software Development Life Cycle (SDLC) process.

* Analysis
* Design
* Implementation
* Testing
* Maintenance

1. What will be the output of the below format strings?
2. >>>'My {1[device]} runs {0.platform}'.format(sys, {device: 'laptop'})

'My laptop runs win32'

1. >>>l = ['P', 'Y', 'T', 'H', 'O', 'N']

>>>'first={0}, last={1}, middle={2}'.format(\*l)

'first=P, last=Y, middle=T'

1. Give the regular expression character classes for the below requirements.

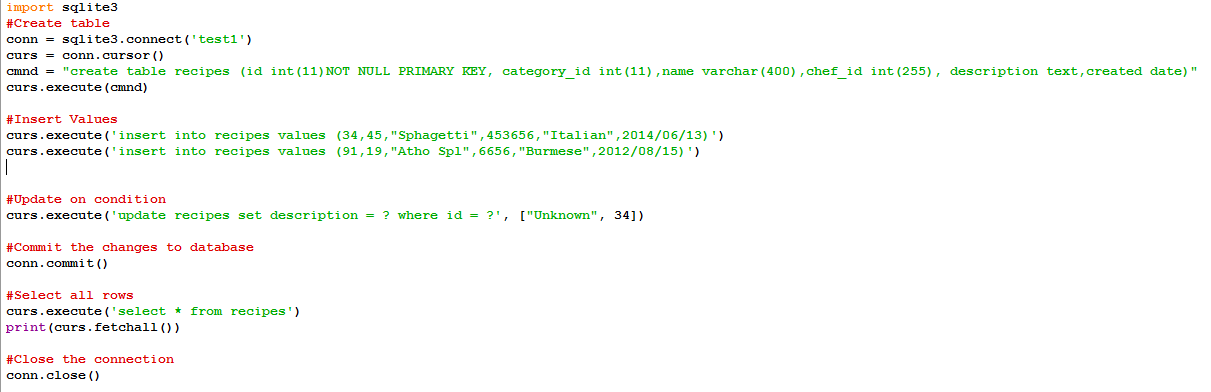
|  |  |
| --- | --- |
| **Requirements** | **Character Class** |
| Match m repetitions of a character “C” | C{m} |
| One or more time of occurrence of a character “C” | C+ |
| Save the match as group | () |
| Non greedy matching | ? |

**PART – B (2\*15 = 30 marks)**

**[ANSER ANY ONE IN EITHER OR PART]**

1. a)Create the below table and execute the insert, update and select statements.



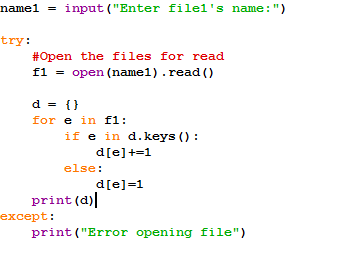


**(OR)**

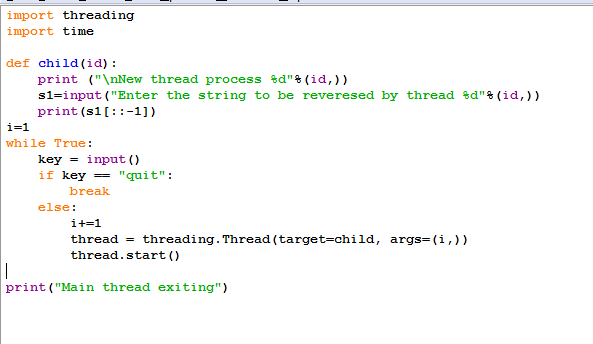
b)Demonstrate a program that prompts the user to enter a file name and counts the number of

occurrences of each letter in the file regardless of case. Ensure that an appropriate error

message is displayed if any problem is encountered by Python interpreter.



1. a)Write a Python program start a thread every time the user enters an input. Each parallel copy of the function simply reverses the user input string and prints the reversed string to standard output.



**(OR)**

b)(i) In a given a file "sample.txt", find the following using regular expression by reading line by line,

When, in disgrace my fortune and men's eyes,

I all alone beweep my outcast state,

And trouble deaf heaven with my bootless cries,

And look upon myself and curse my fatefateandfateandfate fate...!,

**1**. search the word ,"my" and match the index position and return the result as

"my found in position 17 and ends in 18"

2. find the word "fate" in one instance by compiling with findall() function

3. explain the concept of compile in regular expression using above string

(ii)Write a program that match simple Web domain names that begin with “http://www.” and end with a “.com” suffix, e.g., http://www.yahoo.com in the list of websites given as an input.

**8)b)i)**

**1)**

import re

file = open("sample.txt", "r")

for line in file:

if my in line:

s = line.start()

e = line.end()

print ('Found %s in file in range from %d to %d',line.re.pattern,s,e)

2)

import re

file = open("sample.txt", "r")

pattern = 'fate'

for match in re.findall(pattern, file):

print( 'Found "%s"',match)

3)Sample program to demonstrate use of compile

import **re**

*# Pre-compile the patterns*

regexes = [ re.compile(p) **for** p **in** [ 'this',

'that',

]

]

text = 'Does this text match the pattern?'

**for** regex **in** regexes:

**print** ('Looking for "**%s**" in "**%s**" ', % (regex.pattern, text))

**if** regex.search(text):

**print** 'found a match!'

**else**:

**print** 'no match'

b**)ii) Solution 1:**

import re

url = '<p>Hello World</p><a href="http://yahoo.com"> </a><a href="http://google.com"> </a>'

urls = re.findall('http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', url)

print(urls)

**Solution 2:**

import re

string = "This is a link http://www.google.com"

find\_urls\_in\_string = re.compile(regex, re.IGNORECASE)

url = find\_urls\_in\_string.search(string)

if url is not None and url.group(0) is not None:

print("URL parts: " + str(s.groups()))

print("URL" + url.group(0).strip())

**Solution 3:**

myString = "These are the links http://www.google.com and http://yahoo.com"

print re.findall(r'(https?://[^\s]+)', myString)