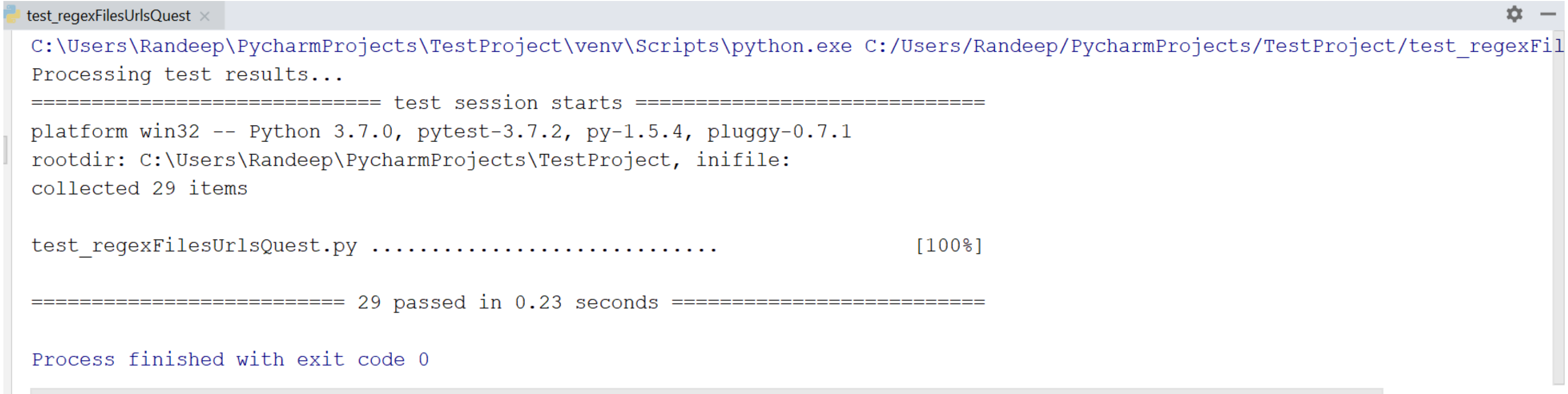
CLASS: CSE7345
NAME: Hanspal, Randeep
SMUID: 47812509
QUEST: regexFilesUrlsQuest

CODE:

```
#Quest: Regex, Files, Urls
import re, pytest, requests

__STUDENT_ID__  = "47812509"
__QUEST_NAME__ = "Quest2"

def count_vowels(mystr):
    #findall the vowels from mystr and return its length
    match=re.findall('[aeiou|AEIOU]', mystr)
    return (len(match))

def is_valid_python_hex(mystr):
    match = re.fullmatch('0x[A-F0-9a-f]+', mystr)
    return match is not None

def has_vowel(mystr):
    match = re.findall('[aeiou|AEIOU]',mystr)
    return len(match)

def is_integer(mystr):
    match = re.fullmatch('-?\d+', mystr)
    return match is not None

def get_extension(mystr):
    #regex to search the extension which includes .(dot)
    match = re.search('\.[\w]*$', mystr)
    if match is not None:
        return (match.group()[1:])
    return 'NONE'

def is_number(mystr):
    #regex to match if number is exact match
    match=re.fullmatch('-?\d{3,}(\.[0-9]{0,})?',mystr)
    return match is not None

def convert_date_format(mystr):
    match = re.match('\d{4}[-]\d{2}[-]\d{2}', mystr)
    if match is not None and match.group() == mystr:
        #split mystr into variable year,mo,day
        year,mo,day = mystr.split('-')
        return (mo + '-' + day + '-' + year)
    return 'NONE'

#File functions
def readFileCountLines(filename):
    fop=open(filename)
    count = 0
    for line in fop:
        #check each line is not blank & contains non-white space
        if re.match('[\S\s]+',line): count+=1
    return(count)

def readFileCountStringOccurrences(filename, stringval):
    fop = open(filename)
    count=0
    for line in fop:
        #search fn to search rollo on each line
        match=re.search(r'rollo',line)
        if match: count+=1
    return(count)

def readFileSumDigitsGreaterThanNumber(filename, number):
    fop = open('pytestFile1.txt')
    count=0
    for line in fop:
        #regex to find number greater than 15
        match=re.search(r'1[6-9]|[2-9]\d',line)
        if match:
            x=int(match.group())
            count+=x
    return(count)

def remove_all_but_alpha(mystr):
    #remove all the special chars(-,9,*) other than alphabets
    match=re.split('[^a-zA-Z]',mystr)
    #use join method to join the string from mystr
    z=[''.join(match[0:(len(match))])]
    return(z[0])

#URL functions
def readurlCountStringOccurrences(urlname, stringval):
    response = requests.get(urlname)
    #regex to findall the stringval to be case insensitive
    match=re.findall(stringval,response.text, re.IGNORECASE)
    return(len(match))
```

```
def readurlCountValidPhoneNumbers(urlname):
    response = requests.get(urlname)
    #match1: find the phone number with -(hyphen) seperator
    match1=re.findall('\d{3}-\d{3}-\d{4}',response.text)
    # match2: find the phone number with total length of 10chars
    match2=re.findall('\d{10}',response.text)
    # match3: find the phone number with .(dot) seperator
    match3=re.findall('\d{3}[.]\d{3}[.]\d{4}',response.text)
    #match4:add all the above 3 match
    match4=(match1,match2,match3)
    return(len(match4))

if __name__  == '__main__':
    print ("To test your code execute: python
test_QuestFilesUrls.py  or on command line execute: pytest ")
```

OUTPUT:



```
C:\Users\Randeep\PycharmProjects\TestProject\venv\Scripts\python.exe C:/Users/Randeep/PycharmProjects/TestProject/test_regexFil
Processing test results...
============================= test session starts =============================
platform win32 -- Python 3.7.0, pytest-3.7.2, py-1.5.4, pluggy-0.7.1
rootdir: C:\Users\Randeep\PycharmProjects\TestProject, inifile:
collected 29 items

test_regexFilesUrlsQuest.py .............................                [100%]

========================== 29 passed in 0.23 seconds ==========================

Process finished with exit code 0
```