|  |  |  |
| --- | --- | --- |
| **الجمهورية العربية السورية** | logo | Syrian Arab Republic |
| **اللاذقية -جامعـــة تشريـــــن** | Lattakia - Tishreen University |
| **كلية الهندسة الكهربائية والميكانيكية**  **قسم هندسة الاتصالات والالكترونيات**  **السنة الخامسة: وظيفة1 برمجة شبكات** | Department of Communication and electrical engineering  5th , Network Programming : Homework No1 |

Name:علي نزار رزوق, Number:844 , Submitted To GitHub:\_\_\_\_\_\_\_

First Network Programming Homework

**Question 1:** Python Basics?

**A-Define a list that contain the names of graduated students” 5 students at least”:**

**Create a program that accept student name and prints if the user is graduated or not.**

Solution steps:

* First we define the list of graduated students.
* Then we should make the user enter a name to search for.(using input)
* Then we should test if the user entered a name that is in the list "graduated", the program should print a sentence with meaning that the student has graduated.
* if the user entered a name that is not in the list "graduated", the program should print a sentence with meaning that the student has not graduated.
* We used "while true" loop to make the ability to redo the search more than one time.

The code:

graduated = ["ahmad", "sami", "haidar", "ali", "mahmoud"]

guess = input("Enter a student name: ")

while True:

success = 0

for i in graduated:

if guess.lower() == i:

success=1

if success==1:

print("Student", guess, " has graduated")

else:

print("Student", guess, " has not graduated")

next\_input = input("Do you want to look up for another student? (yes/no): ")

if next\_input.lower() == "no":

break

elif next\_input.lower() == "yes":

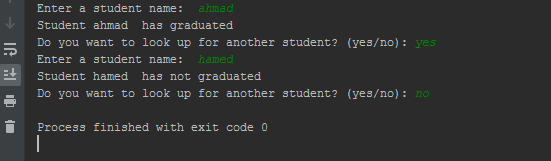
guess = input("Enter a student name: ")

else:

print("Invalid Input")

break

Excution and results:



**B- Generate and print a list of odd numbers from 1 to 1000.**

**Tips: “List Comprehension”**

Solution steps:

* First we define a list that contain all numbers from 1 to 1000.
* Then we define an empty list in order to use it to save the odd numbers later.
* Using for statement we test every number from 1 to 1000 , and if the reminder of its division by 2 equal to 1 so the this number is an odd number.
* If the number is an odd number , we add it to the list "odd\_Numbers" using method append()
* In the end we print the list odd to show the odd numbers from 1 to 1000.

The code:

numbers = list(range(1,1000))

odd\_Numbers = []

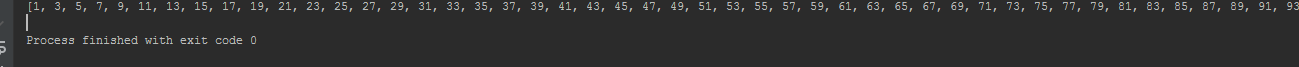
for x in numbers:

if x % 2 ==1:

odd\_Numbers.append(x)

print(odd\_Numbers)

Execution and results:



**C- L=[‘Network’ , ’Math’ , ’Programming’, ‘Physics’ , ‘Music’]**

**In this exercise, you will implement a Python program that reads the items of the previous list and identifies   
 the items that starts with ‘P’ letter, then print it on screen.  
 Tips: using loop, list ‘len ()’ method**

Solution steps:

* First of all, we define a list "L" that contain the names of items .
* Then we print this list before editing.
* Then we test if the name stating with "P" , we add it to a new matrix named "result" using append() method.
* Then we print the list "result".

The code :

L = ['Network', 'Math', 'Programming', 'Physics', 'Music']

check = 'P'

result=[]

print("The items of the original list are : " + str(L))

for ind in range(0,len(L)):

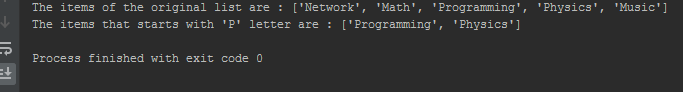
item = L[ind]

if item[0].lower() == check.lower():

result.append(item)

print("The items that starts with 'P' letter are : " + str(result))

Execution and results:



**D: Using Dictionary comprehension, Generate this dictionary d={1:1,2:4,3:9,4:16,5:25,6:36,7:42,8:64,9:81,10:100}**

Solution steps:

* First we define an empty dictionary.
* Then we add the required information to the dictionary (To make it easier, we use "for" loop ).
* Finally we should print the dictionary.

The code:

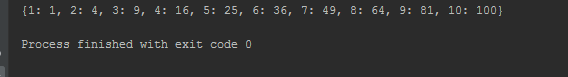
dictionary = {}

for i in range(1,11):

dictionary[i] = i\*i

print(dictionary)

Execution and results:



**Question 2: Convert from decimal to binary**

**Write a Python program that converts a decimal number into its equivalent binary number.**

**The program should start reading the decimal number from the user. Then the binary equivalent number must be calculated. Finally, the program must display the equivalent binary number on the screen.**

**Tips: use empty list to hold binary number, use loop, use % operator, use // operator, use list append method, reverse the list.**

Solution steps:

* First we use the input statement to ask the user to input a decimal number.
* Then we should divide the number by 2 and add the reminder to the list "rem" using method append().
* We repeat the last step until the divide result equal to 0.
* In the end we have to reverse the "rem" list using method reverse().

The code:

n=0

while n < 1:

n = int(input("Enter a Valid Decimal Number: "))

rem = []

while n > 0:

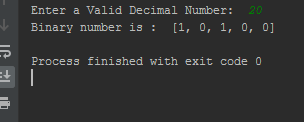
rem.append(n % 2)

n //= 2

rem.reverse()

print("Binary number is : ", rem)

Execution and results:

****

Question 3: Working with Files” Quiz Program”

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file.

Solution steps:

* Open question file "questions.txt"(note that this file contain the questions and its answers, each question followed by its correct answer in the next line . for example: first line contain the first question , and the second line contain the correct answer for the first question , and so on.) to read from it using statement: with open('questions.txt') as f:
* Then read lines and store it in list "Lines".
* Then ask the user to input his name .
* Then we read lines and we have two cases :

1. If the line is odd so this line is a question , so we have to add it to the question list using append() method.
2. If the line is even so this line is an correct answer , so we have to add it to the answer list using append() method.

* Then print each question and ask the user to input his answer.
* Then we have to compare the entered value with the correct value, if they are the same we should increase the result ("numRight") by 1.
* Then print the result.
* In the end we have to save the name and result to the file "result.txt" using this statements :

with open ('result.txt','w') as f:

f.write('Name : ')

f.write(name)

f.write(', grade: ')

f.write(str(numRight))

**note:** I will send the files "question.txt" and "result.txt" in separate files.

The code:

with open('questions.txt') as f:

lines=f.readlines()

count = 0

numRight = 0

answer=[]

user\_Ans=[]

questions=[]

name=input("Enter your name")

for line in lines:

count=count+1

if count % 2 == 1:

questions.append(line.strip().split("\t"))

if count%2==0:

answer.append(line.strip().split("\t"))

for i in range(20):

print("Question",i+1," : ",questions[i])

ans = input("Your answer")

user\_Ans.append(ans)

answers=answer[i]

if ans==answers[0]:

numRight=numRight+1

print("Your grade is",numRight," from 20")

with open ('result.txt','w') as f:

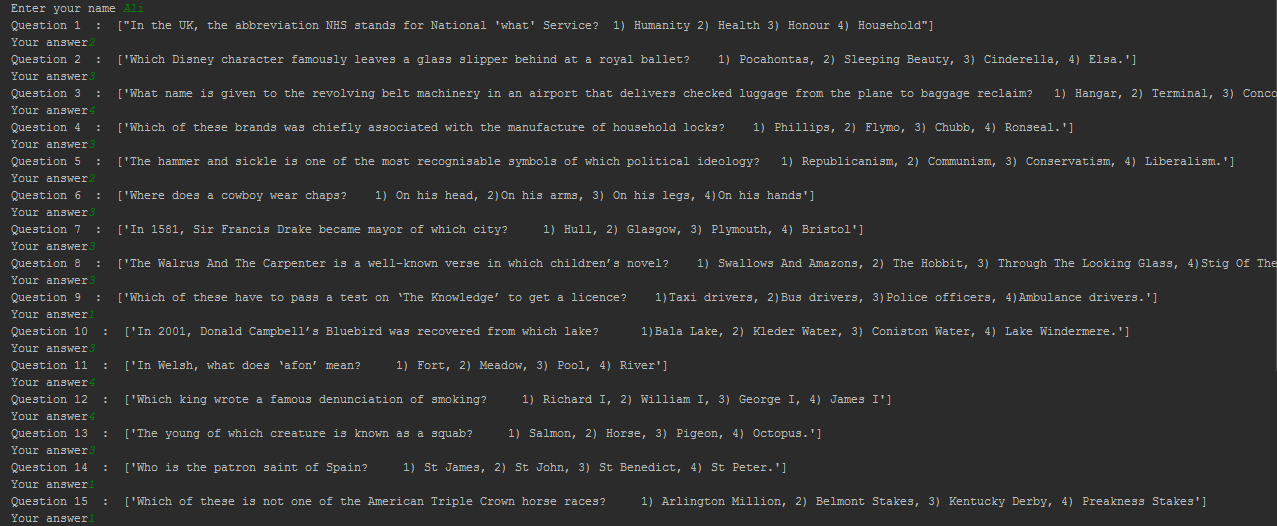
f.write('Name : ')

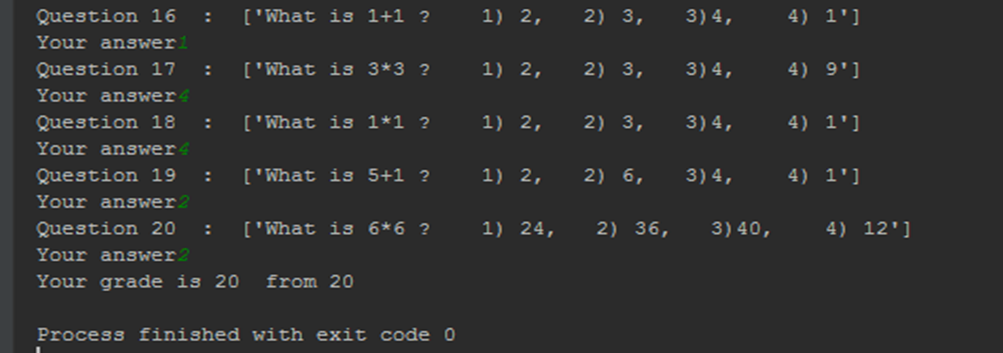
f.write(name)

f.write(', grade: ')

f.write(str(numRight))

Execution and results:



****

**The end**