|  |
| --- |
| **SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY SOFTWARE ENGINEERING DEPARTMENT**    **Fall 2020F**    **Programming Fundamentals (SWE-102) Assignment # 3**    **Semester: I Batch: 2020**  **Max Points: 10**    **Instructions:**   * Attempt all questions * Do not copy or cheat from any one, make your own effort. If any assignment found copied it will straight away be rejected. * All questions solution contains Source Code and Output snapshot using Python Thonny IDE. * There is no maximum page limit. * Assignment should be in MS Word document. The Word file should be submitted by converting it to a single PDF file. * Mention your full name and roll number on the first / front page. The page numbers must be mentioned at the right bottom corner in the whole assignment. * The maximum font size for text is 12 and for heading is 14. The font could be Arial or the Time New Roman |

# Question 1 [2.5 marks]

Write a Python program to take 6 random numbers using ‘Random’ module, print all random numbers and count the number of even and odd numbers from the generated random numbers.

Output looks alike: A: 2 , B:5 , C: 6 , D:9 , E:7 , R:3

Number of even numbers: 2

Number of odd numbers: 4

**CODE:**

|  |
| --- |
| #Abdul Moiz Chishti  #BSE-20F-022  import random  a=[random.randrange(1, 50) for i in range(6)]  xyz=["A:"+str(a[0]),"B:"+str(a[1]),"C:"+str(a[2]),"D:"+str(a[3]),"E:"+str(a[4]),"F:"+str(a[5]),]  print(xyz)  odd = 0  even = 0  for j in a:  if not j % 2:  even+=1  else:  odd+=1  print("number of even numbers :",even)  print("number of odd numbers :",odd) |

**OUTPUT:**

|  |
| --- |
|  |

# Question 2 [2.5 marks]

Write a Python program to make a list by taking 15 inputs from the user, and repeat some inputs value as same. Then delete all repeated elements of the list.

Output looks alike: INPUT:[2,3,3,6,7,8,9,9,11,12,13,17,22,2,22]

OUTPUT: [2,3,6,7,8,9,11,12,13,17,22]

**CODE:**

|  |
| --- |
| #Abdul Moiz Chishti  #Bse-20F-022  i = 15  a = []  while i > 0:  num = input("Enter number: ")  a.append(num)  i = i-1  print (a)  abc = []  for i in a:  if i not in abc:  abc.append(i)  print ("list after removing duplicates : " + str(abc)) |

**OUTPUT:**

|  |
| --- |
|  |

# Question 3 [2.5 marks]

Write a Python program to create a dictionary called groceries with the values, items like "rice", "orange", and "apple juice" etc, take at least 5 groceries items. Perform following task:

* Define a function called “calculate\_bill” that takes one argument food as input. In the function, create a variable total with an initial value of zero. For each item in the food list, add the price of that item to total. In the end, return the total as output.
* Make the following changes to your calculate\_bill function: While you loop through each item of food, only add the price of the item to total if the item's price is greater than 20. Also add 3 more items in your grocery

items and remove 2 less price items in your previous groceries item. Fetch the final output

**CODE:**

|  |
| --- |
| #Abdul Moiz Chishti  #BSE-20F-022  groceries={'rice':3,'wheat':6,'oil':9,'eggs':12,'bread':15}  def calculate\_bill():  total=0  count=0    key\_list = list(groceries.keys())  val\_list = list(groceries.values())  choice= input("Enter the food:")  price = key\_list.index(choice)    a=val\_list[price]  total= total+a  print("Price of ",choice,"=",total)  calculate\_bill()  def calculate\_bil():  print("")  total=sum(groceries.values())  print("Total sum of old list =",total,"\n")  if total >20:  groceries["milk"]=300  groceries["salt"]=100  groceries["sugar"]=200      del groceries["rice"]  del groceries["wheat"]  print("New list=",groceries)  total=sum(groceries.values())  print("\nTotal Sum of new list =",total)  calculate\_bil() |

**OUTPUT:**

|  |
| --- |
|  |

# Question 4

**LOGIN APPLICATION PAGE:**Write a Python program to take the user name and password from the user and compare with the storing data, if the input matches with the stored data so display the home page with multiple options otherwise display invalid data, every user should have two chances to re-enter the data. Use decisions and nested looping. Output looks alike:

\*\*\*\*\*Login Page\*\*\*\*\*

User name: XXXXXX

Password: XXXXX

\*\*\*\*\*\*\*\*\*\*\*\* Home page

\*\*\*\*\*\*\*\*\*

Menu

\*\*\*\*\*\*\*\*\*

Setting

\*\*\*\*\*\*\*\*\*

Contact us

\*\*\*\*\*\*\*\*\*

If data not matches

Re-enter the data? Y/N

**CODE:**

|  |
| --- |
| #Abdul Moiz Chishti  #BSE-20F-022  cred=["admin","123"]  count = 0  while True:  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LOGIN PAGE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  username = input("Username: ")  password = input("Password: ")  count = count + 1  if count == 2:  print("Login Attempt Over")  break    else:  if username == cred[0] and password == cred[1]:  print("Welcome\n")  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HOME PAGE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*SETTING\*\*\*\*\*\*\*\*\*\*\*\n")  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*CONTACT US\*\*\*\*\*\*\*\*\n")  break    else:  print("\nUser Not Found\n") |

**OUTPUT:**

|  |
| --- |
|  |

Page 1 of 1