HADHRAMOUT UNIVERSITY
COLLEGE OF COMPUTERS & INFORMATION TECHNOLOGY
FINAL EXAMINATION

Academic year: *2020-2021*
Day and Date:
Examiner: *Hussien . O . A_Ibaiti*
Time allowed: *2*

Exam Semester : *First*
Level: *Third*
Department: *IT*
Subject: *Data Structure*

(B)

Answer following questions by use one class only :

1) Write function that print your name in reverse way by using Stack concept .

Note:

- User accepted string out side class by object and don't use ***def__ init__*** [2 Marks]
- Print string out side class [2 Marks]

2) Write function to add set of letters to list by using DQ concept .

Note:

- If user enter consonant letters add to rear side **if** enter vowels letters add to front side [4 Marks]
- If user enter number add to front [2 Marks]

3) Write a function to revers the words that are less than 4 from a given list of words . don't use revers function

Example:

lst = ['apple', 'car', 'box', 'window', 'milk', 'one']

output => ['apple', 'rac' , 'xob' ,'window', 'milk','eno']

Note:

- user accept items inside class [1 Marks]
- user sorted items inside class [2 Marks]
- print items out side class [2 Marks]

```
Class DS:
    L=[]
    IL=[]
    def revName(self,s):
        st=""
        self.IL=[]
        for i in s:
            self.L.append(i)
        for j in range(len(self.L)):
            x=self.L.pop()
            self.IL.append(x)
        for i in range(len(self.IL)):
            st1=self.IL.pop()
            st=st1+st
        return st

    def deque(self):
        dequ=[]
        front=rear=-1
        for i in range(3):
            key=input("enter:")
            if front<0:
                if key not in "aOuUiIeEaA" and key not in "0123456789":
                    dequ.append(key)
                else:
                    dequ.insert(0,key)
                front+=1
            else:
                if key not in "aOuUiIeEaA" and key not in "0123456789":
                    dequ.append(key)
                else:
                    dequ.insert(0,key)
                rear+=1
        return dequ
```

22.5

Data Structures - Monthly Test

Name: [redacted] Department: IT A

Q. 1: Put (√) or (×) on the appropriate sentences, with correct the false ones:

1- Stacks are classified into primitive and non-primitive. ×

2- If the elements are connected in a linear sequence memory locations, it called linear data structure. ✓

3- Circle queue is an example of non-linear data structure. ×
Graph

4- An Queue can be accessed only at one of its ends called (top), for either adding or deleting. ×
Stacks

5- Adding an element to the Queue is called enqueue, ✓

6- Correctness means every step of the algorithm should be clear and well defined. ×
definiteness

7- In the queue the operation Front is return the last element in it. ×
First

Q. 2: Compare (قارن) between stack and queue in short:

Comparison Points (نقاط المقارنة)	Stack	Queue
Principle المبدأ	(First In Last out) or (Last In First out)	(First in First out) (Last in Last out)
Adding الإضافة	Push() by one end (ToP)	by Rear
Deleting الحذف	by one end (ToP), PoP()	by Front PoP()
First element أول عنصر	—	L.Front()
Last element آخر عنصر	L.ToP()	L.Rear()

PoP(), Push(), Front, Rear

Q.3: Mention (اذكر) two applications of circle queue?

1- trafic systems.
2- CPU scaduial
3- memory management

Q. 4: Write a recursive function to find the Greatest Common Divisor (GCD), where, the function takes 2 parameters as input, the following recursive definition determines the GCD of x and y:

$$\gcd(x,y) = \begin{cases} x & \text{if } y = 0 \\ \gcd(y, x \% y) & \text{if } y \neq 0 \end{cases}$$

بالتوفيق

```
def Gcd(x,y):
    If y==0:
        return x
    else:
        return Gcd(y, x%y)
```

```
Print(Gcd(12,4))
```

Ex

Gcd(12, 4)
(x, y)

↳ 12/4 = 3 % => 0

↓ y ≠ 0

Gcd(4, 0)
x%x = 0

↳ Then y = 0

then return x

[redacted]

G12 Assignment (A)

Write a program to revers the words that are less than 4 from a given list of words . don't use revers function

Example:

lst = ['apple', 'car', 'box', 'window', 'milk', 'one']

output => ['apple', 'rac' , 'xob' ,'window', 'milk','eno']

Note (

1) User specific list size out side class by object
2) user accept items inside class
3) create function to search about words that interact with it from inside class
4) print items out side class

)

3/5

G2 Assign

(A) [redacted]

Write a Python program that takes a list from user with unique odd items

Example:

lst = [1, 2, 2, 3, 3, 3, 4, 5,5]

output => [1, 2,2, 3, 4, 5]

Note (

1) Use stack concept
2) use class concept
3) print/list outside class

)

-> 3/5

```
Class stack:
    L=[ 1, 2, 2, 3, 3, 3, 4,5 ,5]
    top=len(L)-1
    def POP(self,x):
        global top
        top-=1
        return self.POP(x)
    def Push(self,i):
        global top, newl
        newl.append(i)
    def check(self):
        newl=[]
        for i in range(len(self.L)):
            if self.L[i]%2!=0 and self.L[i] not in new
                newl.append(self.L[i])
            elif self.L[i]%2==0:
                newl.append(self.L[i])
        return newl
oop=stack()
Print(oop.check())
```

not stack concept