

PYTHON CB13

Solutions of Exercises III

1. Cylinder surface and area:

```
import math

while True:

    choice=input("Do you like to find a cylinder's volume and surface area? Choose:'y' or 'n'").lower()

    if choice=='n':

        print("Finishing....")

        break

    elif choice=='y':

        while True:

            try:

                height = float(input("Height of cylinder: "))

            except ValueError:

                print("Something wrong! Try again!")

                continue

            else:

                break

        while True:

            try:

                radian = float(input("Radius of cylinder:"))

            except ValueError:

                print("Something wrong! Try again!")

                continue

            else:

                break

        volume = math.pi * radian * radian * height

        sur_area = ((2*math.pi*radian) * height) + ((math.pi*radian**2)*2)
```

```
print("Volume is: ", volume)

print("Surface Area is: ", sur_area)

else:

    print("Wrong choice. Please try again")
```

2. The queue

Άσκηση Ουρά δύο άκρων

dqueue = [] # Δημιουργώ τη λίστα (ουρά) η οποία στην αρχή είναι κενή

inp = None

while True:

```
inp = input("Input can be: data / 0data / r / 0r / q : ")
```

```
if inp == 'q': # Ελέγχω για είσοδο q
```

```
    break
```

```
elif 'r' in inp:      # Ελέγχω αν περιλαμβάνεται το r
```

```
    if len(dqueue)>0:  # Ελέγχω αν η ουρά είναι κενή
```

```
        if inp[0]=='0': # Αν προηγείται 0
```

```
            dqueue.pop(0) # Αφαιρώ το πρώτο στοιχείο από την ουρά με pop(0)
```

```
        else:
```

```
            dqueue.pop() # Αλλιώς αφαιρώ το τελευταίο στοιχείο από την ουρά απλά με pop()
```

```
    else:
```

```
        print("Empty queue")
```

```
else:
```

```
    if inp[0]=='0': # Ελέγχω το πρώτο στοιχείο αν είναι 0
```

```
        dqueue.insert(0,inp) # Εισάγω το δεδομένο στη θέση 0, δηλ. στην αρχή της ουράς
```

```
    else:
```

```
        dqueue.append(inp) # Αλλιώς εισάγω το δεδομένο στο τέλος της ουράς
```

```
print(dqueue)
```

```
# ΣΧΟΛΙΑ
```

```
# Στη λύση που δίνω χειρίζομαι τα δεδομένα ως χαρακτήρες
```

```
# Αν χρειάζεται μπορώ να καλώ την int() ή float() και να τα μετατρέπω σε ακέραιες ή πραγματικές τιμές
```

3. Lions

```
lionsToday = 23000
```

```
reductionPercentage = 2/100
```

```
yearsFromToday = 0
```

```
while lionsToday > 6000:
```

```
    lionsToday = lionsToday - lionsToday * reductionPercentage
```

```
    yearsFromToday = yearsFromToday + 1
```

```
print("Today lions are 23000")
```

```
print("After " + str(yearsFromToday) + " years lions will be one of endangered species")
```

4. Stats

```
import random as rn
```

```
import math
```

```
def stats(lista):
```

```
    m = sum(lista)/len(lista)    # Μέσος όρος, το m είναι τοπικό όνομα
```

```
    suma=0
```

```
    for x in lista:
```

```
        suma += (x-m)**2        # Άθροισμα τετραγώνων
```

```
    s = math.sqrt(suma/(len(lista)-1)) # Τυπική απόκλιση, το s επίσης είναι τοπικό όνομα
```

```
    return m,s                  # Επιστρέφει στο κύριο πρόγραμμα τα m & s
```

```
def main():
```

```
    alist=[]
```

```

for i in range(100):
    alist.append(rn.randint(1,20))

    #alist=[rn.randint(1,20)for i in range(100)] # Δημιουργία με list comprehension, η alist ανήκει στην καθολική
    #εμβέλεια

    mo, sd = stats(alist)          # Κληση της stats με παράμετρο την alist

    print(alist)

    print('Μέσος όρος = {:.3f}, Τυπική απόκλιση = {:.3f}'.format(mo,sd))

if __name__=='__main__':
    main()

```

5. Date and Time information Module

```

import time

import datetime

print("Time in seconds since the epoch: %s" %time.time())

print("Current date and time: " , datetime.datetime.now())

print("Alternate date and time: " ,datetime.datetime.now().strftime("%y-%m-%d-%H-%M"))

print("Current year: " , datetime.date.today().strftime("%Y"))

print("Month of year: " , datetime.date.today().strftime("%B"))

print("Week number of the year: " , datetime.date.today().strftime("%W"))

print("Weekday of the week: " , datetime.date.today().strftime("%w"))

print("Day of year: " , datetime.date.today().strftime("%j"))

print("Day of the month : " , datetime.date.today().strftime("%d"))

print("Day of week: " , datetime.date.today().strftime("%A"))

print()

```

6. Five days from Current

```
from datetime import date, timedelta  
dt = date.today() - timedelta(5)  
print('Current Date :',date.today())  
print('5 days before Current Date :',dt)
```

7. Εμβαδόν ορθογωνίου

```
def AskForUserInputs(sideName):  
    while True:  
        try:  
            s = float(input("Dwse "+ sideName))  
        except ValueError:  
            print("Not valid float! Please try again")  
        else:  
            break  
    return s
```

```
def rectangleAcreage(side1,side2):  
    acreage = side1 * side2  
    return acreage
```

```
def EmvadoEmfanisi():  
    plevra1 = AskForUserInputs("Ypsos")  
    plevra2 = AskForUserInputs("Mhkos")  
    print(rectangleAcreage(plevra1,plevra2))
```

```
EmvadoEmfanisi()
```

8. 1^η Ενδεικτική Λύση για την ακολουθία Fibonacci (with while)

```
def recursive_fib(n):  
    if n <= 1:  
        return n  
    else:  
        return(recursive_fib(n-1) + recursive_fib(n-2))
```

```
def main():  
    n=int(input("Give a number"))  
    if n<= 0:  
        print("Plese enter a positive integer")  
    else:  
        print("Fibonacci sequence:")  
        for i in range(n):  
            print(recursive_fib(i))
```

```
if __name__=='__main__':  
    main()
```

2^η Ενδεικτική Λύση για την ακολουθία Fibonacci (with recursion)

```
n = int(input("Enter the value of 'n': "))  
a = 0  
b = 1  
sum = 0  
count = 1  
print("Fibonacci Series: ", end = " ")  
while(count <= n):  
    print(sum, end = " ")  
    count += 1  
    a = b  
    b = sum
```

```
sum = a + b
```

9. Lines program

```
lines = []
```

```
while True:
```

```
    l = input("Give a new line")
```

```
    if l:
```

```
        lines.append(l.lower())
```

```
    else:
```

```
        break
```

```
for l in lines:
```

```
    print(l)
```

10. List with animals

```
list1 = ["goat", "bufallo", "sheep", "dog", "cat"]
```

```
list1.sort()
```

```
print(list1)
```

```
list1.reverse()
```

```
print(list1)
```

```
for pet in list1:
```

```
    print(pet)
```

11. Tickets module

```
import random
```

```
myAge = random.randint(1,100)
```

```
print("The age that has been selected from the system is: ",myAge)
```

```
if(myAge > 16):
```

```
    print("Please purchase the adult ticket")
```

```
else:
```

```
    print("Please purchase the child ticket")
```