==
(2) Briefly describe how a python program is executed - give two main steps in it.

Answer:
Python program is first compiled into bytecode by the compiler
Bytecode is then executed by the python virtual machine

==
(4) Write a function **integrate** that computes the integral of a given single-variable function (passed to it as a parameter) from x1 to x2. The increment for x values for computing the integral numerically is also given as a parameter.

Ans:

```
def integrate(f, x1, x2, d):
        x = x1
        sum = 0
        while x < x2:
                sum += ((f(x)+f(x+d))/2)*d
                x += d
        return sum
```

==

(4) Two balls of radius r1 and r2 are to be made from two different metals. The price for constructing each ball varies non-linearly with the volume. Assume that two functions cost1(r) and cost2(r) that compute the cost of the two balls respectively are given to you. Write a program to find the radius of the two balls which will minimize the overall cost of building these two balls (let's call them **r1opt** and **r2opt**), and the cost, subject to the condition that each of them is at least 10 cm radius, and that together their radius should be less than 40 cm. Assume that the cost is always lesser than 10000. Given the code skeleton below, you have to write the code for the main program.

```
def cost1(r):
# returns a float value
def cost2( r):
# returns a float value

# main program
min = 10000 # initializing value of the min cost
# Your code goes here




print(min, r1opt, r2opt)
```

Ans:

```
min = 10000
for r1 in range(10, 30):
        for r2 in range(10, (40-r1)):
                cost = cost1(r1) + cost2 (r2)
                if cost < min:
                        min=cost
                        r1opt, r2opt = r1, r2
print(min, r1opt, r2opt)
```

==

(2) A program to determine if a given list of integers is a palindrome – i.e. reverse is same as the original. You have to fill in ?? (you can only put statements in the loop).

```
lst = [ …. ]
palin = True
for i in ??

        ??

print ('Is palindrome: ', palin)
```

Ans:

```
for i in range(len(l)//2):
   palin = palin and (l[i] == l[-i-1])
```

or

```
for i in range(len(l)//2):
   if (l[i] != l[-i-1]):
     palin = False
     break
```

==

(2) For a given a list **lst** and an element **elt**, write a program to create a list of all indices of the element elt. For example if lst [4, 2, 3, 4, 5, 6, 4], for elt=4, ans is [0, 3, 6]; for elt=9, the ans is []. Fill the missing part of the this code:

```
elt = input()  # takes elt as input, e.g. 4
lst = [ <num>, <num>, …] # eg. [4, 2, 3, 4, 5, 6, 4]  - for elt=4, ans is [0, 3, 6]
indices = []
# your code goes here




print(indices)
```

Ans:

```
for i in range(len(l)):
   if lst[i] == elt:
      indices.append(i)
```

==

(2) Given below is a program to count the total number of times an elt is there in a list of lists. Please fill in the code:

```
elt = input()
lst = [ …] # e.g. [[1,2,3], [1, 1, 2], [3,2, 1]]
count = 0
        # Your code goes here – no more than 3 lines


print(count)

for row in lst:
        count += row.count(elt)
```

or

```
for row in lst:
        for e in row:
                if e == elt:
                        count += 1
```

==

(2) Given two lists of lst1 and lst2, create a list of common items using list comprehension.
// give partial marks if done correctly but without using list comprehension

Ans:

```
l = [elt for elt in lst1 if elt in lst2]
```

Or - l = [i for i in x for j in y if i==j]

==

(2) Given a string s which contains integers separated by comma (e.g. “23, 25, 56”), write a single line of code to create a list **lst** which is the list of integers (i.e. [23, 25, 26]) from this string. (Cannot use map or other concepts not covered in class.)

```
lst = [int(e) for e in s.split(",")]
```

or

```
lst=[int(e) for e in s.split(", ")] //This one has a space after the comma. This is also acceptable.
```

==

(2) Given two lists **wt** containing weight (in kgs) and **ht** containing height (in meters) of students in a class. Create a list **bmi** containing BMI of these students (BMI is wt in kgs divided by height-square).

Ans:
(a) bmi = [wt[i]/ht[i]**2 for i in range(len(wt))]

(b) Answer may be a loop - this should also be treated as correct

==

(3, 3) Write a function **reverse(n)** that takes an integer as input and returns the reverse of it (i.e. if the input is 3769 the return value is 9673). Provide two solutions to this: (i) one without using strings or lists, (ii) and one in which you can use strings/lists:

Answer i:

```
nn = 0
while n > 0:
   nn = nn*10 + n%10
   n = n//10
return(nn)
```

Answer ii-A:

```
l = [d for d in str(n)]
l.reverse()
return( int(“”.join(l)))
```

Ans ii-B.

```
s = str(n)
s = s[::-1]
return(int(s))
```

==

(2) Given a word (w) and a sentence (s) – write one expression that will compute how many times the word w occurs in s.

Answer: (s.split()).count(w) //assuming there is no full stop at the end
or
(s[:-1:].split()).count(w) //assuming that the last word is accompanied by a full stop

==

(2) What will be the output of this program:

```
l1 = [1, 3]
l2 = [2, 4]
l = l1*2 + l2
l.pop()
del(l[3])
l.insert(l.index(2), 7)
print(l)

l2.insert(len(l2),6)
l2.pop(len(l2))
print(l2)
```

Ans:

[1, 3, 1, 7, 2] # for first print
Index error # pop(len(l2)) will give this error

==

(2) What will be the output of this program, if the input given is: 4 6 9 15 5 (a return after every input)

```
sum2, sum3 = 0, 0
while True:
   x = int(input())
   if x%2 == 0:
      sum2 += x
   elif x%3 == 0:
      sum3 += x
   else:
      break
print(sum2, sum3)
```

Ans:
10, 24

==

(1 each) In the following questions, you have to provide the definition statement for a function (name of the function given in **bold** in the question) with some comments explaining the type of the parameters, and the type of the return value. You don't have to write the code for the function, but assume that function has no input statements and cannot accesses global variables. So, for example for: "Write a function to **count** how many times the number exists in the list". The answer to this can be::

```
def count(l, n):
# l is a list of integers
# n is an int
```

returns int

Think of some which will also have default values. Look at some standard functions in python – and use them.

a. A function to determine one **root** of a polynomial.

   (Ans: def root(l): # l is the list of coefficients of the polynomial – the length of l specifies the degree of the polynomial; returns a float)

b. A function that prints what type of **triangle** it is, given the sides of the triangle.

   (Ans: def triangle(a, b, c): ; a, b, c: float; returns None

c. Write a function that will **insert** a digit at a position i (i.e. after i number of digits from the right) in a given number; If position is not specified, the digit is added to the end of the number. E.g. inserting 5 at position 2 in 2389 will result in 23589.

   (Ans: def insert(n,d, i=0); n, d, i: integer (some students may give default value as 1, which is fine); return is integer).

   //one can interpret the end of the number as the units digit, so in that case i=len(str(n)) is the default index

d. A function to compute the **product** of two univariate functions at two given points.

   (Ans: def product (f, g, x, y): # f and g are functions; x,y: float; returns float)

e. A function to determine the ability to **purchase** a house given the savings amount, a broker fee, the price of the house, and a standard registration tax of 10%.

   (Ans: def purchase(savings, fee, price, tax=0.1): bool #savings, fee and price are float; returns boolean)

   //In this question, students may think that since the registration rate is standard, it is invariant, hence no need to include it as a parameter as the user should not be able to change it. So, we should accept a function definition without the rate as a parameter, in addition to the given solution

f. Ideal gas equation is given by PV=nRT, where P represents the pressure(in atm), V represents the Volume(in litre), N represents the number of moles of the ideal gas, whose default value is 1, T represents the temperature(in Kelvin), R represents the universal gas constant - its value is 0.0821 unless some other value is specified. Write a function **find_temperature** that computes the temperature.

   (Ans: def find_temperature(P, V, n=1, R=0.0821): #P, V: Float # Returns float)