Question 19

q. arguments passed into function at the function call

a. Actual Parameters

Question 21

q. way to change how a character is interpreted

a. escaping (\)

Question 22

q. L[4]

a. t

Question 26

q. L[:2]

a. Is

Question 32

q. T[:]

a. (1,2,3)

Question 33

q. S+T

a. Error

Question 35

q. What will the code below return for parameter "Rutabaga"?

```
def changeToQuestion(myStr):
     newStr =""
     for i in range(len(myStr)):
          if i%2 == 0:
               newStr = newStr + '?'
          else:
               newStr =newStr + myStr[i]
     return newStr
```

a. ?u?a?a?a

Question 41

q. Write a function called minmax that accepts a list of numbers and returns the minimum and maximum elements of that list. Do not use built-in functions for min or max, but rather write your own.

a.
```
def minmax(list1):
    max=list1[0]
    min=list1[1]
    for i in list1:
        if i > max:
            max=i
    for i in list1:
        if i<min:
            min=i
    return min , max
```

Question 42

q. Write a function called factors that accepts an integer parameter n that computes and returns a list of all the factors of that number, include 1 and n (e.g. factors(6) would return [1, 2, 3, 6]).

a.
```
def factors(n):
    list=[]
    for i in range(1,n+1):
        if n%i == 0:
            list.append(i)
    return list
```

Question 43

q. Write a code fragment to get a number, n, from the user and create a list named `myList` with that many elements. Each element should be a list of n elements counting up in a single sequence. For example, when the user inputs 3, your code should produce:

```
myList = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Your solution should use nested loops.

a.
```
n=int(input("Enter a number: "))
myList=[]
count=1
for i in range(n):
    list2=[]
    for j in range(n):
        list2.append(count)
        count+=1
    myList.append(list2)
```

Question 44

q. Write a program that reads lines from a file named "input.txt" and adds each line that begins with 'A' to the end of a possibly already existing file named "output.txt". For example, if the file "output.txt" did not already exist, after the program processed the file "input.txt", the file "output.txt" would be as below.

Input.txt: Hey yo, I'm just like my country
I'm young, scrappy and hungry
And I'm not throwin' away my shot

Output.txt: And I'm not throwin' away my shot

answer.

```
myInputFile = open(“Input.txt”,”r”)
myInputFile = myInputFile.readlines()
myOutputFile = open(“Output.txt”,”w”)
for eachline in myInputFile:
        if eachline[0]==”A”
                myOutputFile.write(eachline)
myOutputFile.close()
```