**Question# 1:**

Write a program to solve a classic ancient Chinese puzzle: We count 35 heads and 94 legs among the chickens and rabbits in a farm. How many rabbits and how many chickens do we have? Hint: Use for loop to iterate all possible solutions

**Solution:**

heads = 35  
legs = 94  
res = **False  
  
for** r **in** range(1, heads + 1):  
 c = heads - r  
 **if** (2 \* c + 4 \* r) == legs:  
 print(**"There are"**, c, **"Chickens and"**, r, **"Rabbits"**)  
 res = **True  
 break  
  
if not** res:  
 print(**"No solution found"**)

**Result:**



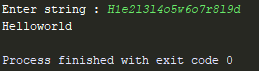
**Question 2:**

Please write a program which accepts a string from console and print the characters that have even indexes. Example: If the following string is given as input to the program: H1e2l3l4o5w6o7r8l9d Then, the output of the program should be: Helloworld Hints: Use list[::2] to iterate a list by step 2.

**Solution:**

inp = input(**"Enter string : "**)  
final=**""  
  
for** ch **in** inp[::2]:  
 final += ch  
  
print(final)

**Result:**



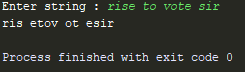
**Question 3:**

Please write a program which accepts a string from console and print it in reverse order. Example: If the following string is given as input to the program: rise to vote sir Then, the output of the program should be: risetovotesir Hints: Use list[::-1] to iterate a list in a reverse order.

**Solution:**

inp = input(**"Enter string : "**)  
final=**""  
  
for** ch **in** inp[::-1]:  
 final += ch  
  
print(final)

**Result:**



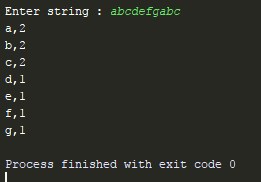
**Question 4:**

Please write a program which count and print the numbers of each character in a string input by console. Example: If the following string is given as input to the program: abcdefgabc Then, the output of the program should be: a,2 c,2 b,2 e,1 d,1 g,1 f,1 Hints: Use dict to store key/value pairs. Use dict.get() method to lookup a key with default value.

**Solution:**

inp = input(**"Enter string : "**)  
char\_dict = {}  
  
**for** ch **in** inp:  
 **if** ch **not in** char\_dict:  
 char\_dict[ch] = 1  
 **elif** ch **in** char\_dict:  
 char\_dict[ch] += 1  
  
**for** key, value **in** char\_dict.items():  
 print(key+**","**+str(value))

**Result:**



**Question 5:**

Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default. Hints: To override a method in super class, we can define a method with the same name in the super class.

**Solution:**

**class** Shape():  
  
 **def** \_\_init\_\_(self):  
 self.area = 0  
  
 **def** get\_area(self):  
 **return** self.area  
  
  
**class** Square(Shape):  
  
 **def** \_\_init\_\_(self, length):  
 Shape.\_\_init\_\_(self)  
 self.length = length  
  
 **def** get\_area(self):  
 **return** self.length\*self.length  
  
square = Square(5)  
square\_area = square.get\_area()  
print(**"Area of square with length "** + str(square.length) + **" is "** + str(square\_area))

**Result:**



**Question 6:**

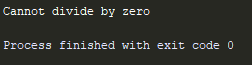
Write a function to compute 5/0 and use try/except to catch the exceptions.

Hints: Use try/except to catch exceptions.

**Solution:**

**try**:  
 res = 5/0  
**except** ZeroDivisionError:  
 print(**"Cannot divide by zero"**)

**Result:**



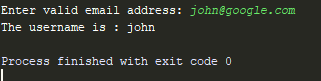
**Question 7:**

Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only. Example: If the following email address is given as input to the program: john@google.com Then, the output of the program should be: john In case of input data being supplied to the question, it should be assumed to be a console input. Hints: Use \w to match letters.

**Solution:**

**import** re  
inp = input(**"Enter valid email address: "**)  
**try**:  
 print(**"The username is : "** + re.search(**"^\w+"**, inp).group(0))  
**except**:  
 print(**"Email address is not valid. Kindly enter valid email address."**)

**Result:**



**Question 8:**

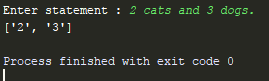
Write a program which accepts a sequence of words separated by whitespace as input to print the words composed of digits only. Example: If the following words is given as input to the program: 2 cats and 3 dogs. Then, the output of the program should be: ['2', '3'] In case of input data being supplied to the question, it should be assumed to be a console input.

Hints: Use re.findall() to find all substring using regex

**Solution:**

**import** re  
  
inp = input(**"Enter statement : "**)  
match = re.findall(**r'[0-9]'**, inp)  
print(match)

**Result:**



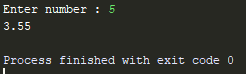
**Question 9:**

Write a program to compute 1/2+2/3+3/4+...+n/n+1 with a given n input by console (n>0). Example: If the following n is given as input to the program: 5 Then, the output of the program should be: 3.55 In case of input data being supplied to the question, it should be assumed to be a console input. Hints: Use float() to convert an integer to a float

**Solution:**

inp = int(input(**"Enter number : "**))  
final = 0  
  
**for** n **in** range(1, inp+1):  
 final += float(n/(n+1))  
  
print(**"%.2f"** % final)

**Result:**



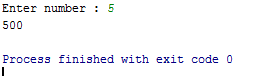
**Question 10:**

Write a program to compute: f(n)=f(n-1)+100 when n>0 and f(0)=1 with a given n input by console (n>0). Example: If the following n is given as input to the program: 5 Then, the output of the program should be: 500 In case of input data being supplied to the question, it should be assumed to be a console input. Hints: We can define recursive function in Python.

**Solution:**

inp = int(input(**"Enter number : "**))  
  
  
**def** compute\_result(number):  
 **if** (number == 0):  
 **return** 0  
 **elif** (number > 0):  
 **return** compute\_result(number-1)+100  
  
print(compute\_result(inp))

**Result:**



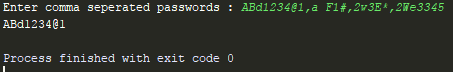
Question 11:

A website requires the users to input username and password to register. Write a program to check the validity of password input by users. Following are the criteria for checking the password: 1. At least 1 letter between [a-z] 2. At least 1 number between [0-9] 1. At least 1 letter between [A-Z] 3. At least 1 character from [$#@] 4. Minimum length of transaction password: 6 5. Maximum length of transaction password: 12 Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. Example If the following passwords are given as input to the program: ABd1234@1,a F1#,2w3E\*,2We3345 Then, the output of the program should be: ABd1234@1 Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

**Solution:**

**import** re  
*#inp = "ABd1234@1,a F1#,2w3E\*,2We3345"*inp = input(**"Enter comma seperated passwords : "**)  
  
**for** password **in** inp.split(**','**):  
 **if**(len(password) >= 6 **and** len(password) <= 12):  
 **if**(re.search(**r"[A-Z]"**, password)):  
 **if** (re.search(**r"[a-z]"**, password)):  
 **if** (re.search(**r"[0-9]"**, password)):  
 **if**(re.search(**r"[$#@]"**, password)):  
 print(password)

**Result:**



**Question 12:**

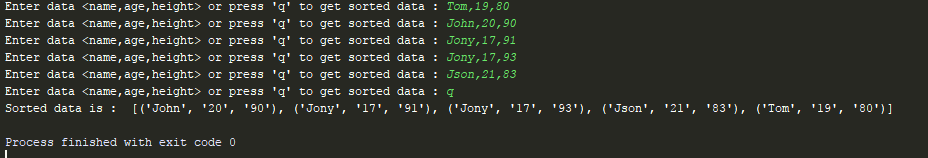
You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is: 1: Sort based on name; 2: Then sort based on age; 3: Then sort by score. The priority is that name > age > score. If the following tuples are given as input to the program: Tom,19,80 John,20,90

Jony,17,91 Jony,17,93 Json,21,85 Then, the output of the program should be: [('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')] Hints: In case of input data being supplied to the question, it should be assumed to be a console input. We use itemgetter to enable multiple sort keys.

**Solution:**

**from** operator **import** itemgetter  
inp = **''**data\_list = []  
data\_tuple = ()  
**while True**:  
 inp = input(**"Enter data <name,age,height> or press 'q' to get sorted data : "**)  
 **if** (inp == **'q'**):  
 **break** data = inp.split(**','**)  
 name = data[0] **if** data[0] **else '0'** age = data[1] **if** data[1] **else** 0  
 height = data[2] **if** data[2] **else** 0  
 data\_tuple = (name, age, height)  
 data\_list.append(data\_tuple)  
  
  
print(**"Sorted data is : "**, sorted(data\_list, key=itemgetter(0, 1, 2)))

**Result:**



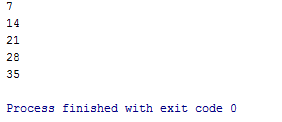
**Question 13:**

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n. Hints: Consider use yield

**Solution:**

**class** gen\_class():  
  
 **def** \_\_init\_\_(self, number):  
 self.num = number  
  
 **def** div\_by\_7(self):  
 **for** n **in** range(1, self.num+1):  
 **if** n % 7 == 0:  
 **yield** n  
  
obj = gen\_class(40)  
**for** x **in** obj.div\_by\_7():  
 print(x)

**Result:**



**Question 14:**

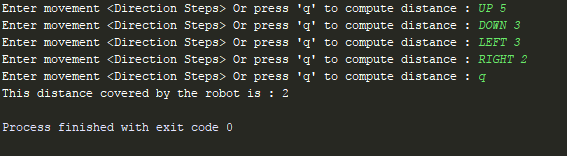
A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps. The trace of robot movement is shown as the following: UP 5 DOWN 3 LEFT 3 RIGHT 2 ¡ The numbers after the direction are steps. Please write a program to compute the distance from current position after a sequence of movement and original point. If the distance is a float, then just print the nearest integer. Example: If the following tuples are given as input to the program: UP 5 DOWN 3 LEFT 3 RIGHT 2 Then, the output of the program should be: 2 Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

**Solution:**

**import** math  
pos = [0,0] *# [x,y]*move1 = **''  
  
while** (move1 != **'q'**):  
 move1 = input(**"Enter movement <Direction Steps> Or press 'q' to compute distance : "**)  
 move = move1.split(**" "**)  
 **if** move[0] == **"UP"**:  
 pos[1] += int(move[1])  
 **elif** move[0] == **"DOWN"**:  
 pos[1] -= int(move[1])  
 **elif** move[0] == **"RIGHT"**:  
 pos[0] += int(move[1])  
 **elif** move[0] == **"LEFT"**:  
 pos[0] -= int(move[1])  
  
  
distance = math.fabs(math.sqrt(pos[0]\*\*2 + pos[1]\*\*2))  
print(**"This distance covered by the robot is "** + str(int(distance)))

**Result:**



**Question 15:**

Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. Suppose the following input is supplied to the program: New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3. Then, the output should be: 2:2 3.:1 3?:1 New:1 Python:5 Read:1 and:1 between:1 choosing:1 or:2 to:1 Hints In case of input data being supplied to the question, it should be assumed to be a console input.

**Solution:**

inp = input(**'Enter statement : '**)  
words = {}  
  
**for** word **in** inp.split(**' '**):  
 **if** word **in** words:  
 words[word] += 1  
 **else**:  
 words[word] = 1  
  
**for** key, value **in** sorted(words.items()):  
 print(key+**":"**+str(value))

**Result:**

