

Join GitHub today

Dismiss

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Branch: master ▾

Python-programming-exercises / 100+ Python challenging programming exercises.txt

Find file

Copy path

 zhiwehu Update 100+ Python challenging programming exercises.txt

22bb9b3 on Apr 27, 2018

2 contributors  

2376 lines (1583 sloc) 50.1 KB

```

1 100+ Python challenging programming exercises
2
3 1.      Level description
4 Level  Description
5 Level 1 Beginner means someone who has just gone through an introductory Python course. He can solve some problems with 1 or 2 Python class
6 Level 2 Intermediate means someone who has just learned Python, but already has a relatively strong programming background from before. He
7 Level 3 Advanced. He should use Python to solve more complex problem using more rich libraries functions and data structures and algorithms
8
9 2.      Problem template
10
11 #-----#
12 Question
13 Hints
14 Solution
15
16 3.      Questions
17
18 #-----#
19 Question 1
20 Level 1
21
22 Question:
23 Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5,
24 between 2000 and 3200 (both included).
25 The numbers obtained should be printed in a comma-separated sequence on a single line.
26
27 Hints:
28 Consider use range(begin, end) method
29
30 Solution:
31 l=[]
32 for i in range(2000, 3201):
33     if (i%7==0) and (i%5!=0):
34         l.append(str(i))
35
36 print ','.join(l)
37 #-----#
38
39 #-----#
40 Question 2
41 Level 1
42
43 Question:
44 Write a program which can compute the factorial of a given numbers.
45 The results should be printed in a comma-separated sequence on a single line.

```

```

46 Suppose the following input is supplied to the program:
47 8
48 Then, the output should be:
49 40320
50
51 Hints:
52 In case of input data being supplied to the question, it should be assumed to be a console input.
53
54 Solution:
55 def fact(x):
56     if x == 0:
57         return 1
58     return x * fact(x - 1)
59
60 x=int(raw_input())
61 print fact(x)
62 #-----#
63
64 #-----#
65 Question 3
66 Level 1
67
68 Question:
69 With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 a
70 Suppose the following input is supplied to the program:
71 8
72 Then, the output should be:
73 {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
74
75 Hints:
76 In case of input data being supplied to the question, it should be assumed to be a console input.
77 Consider use dict()
78
79 Solution:
80 n=int(raw_input())
81 d=dict()
82 for i in range(1,n+1):
83     d[i]=i*i
84
85 print d
86 #-----#
87
88 #-----#
89 Question 4
90 Level 1
91
92 Question:
93 Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every numbe
94 Suppose the following input is supplied to the program:
95 34,67,55,33,12,98
96 Then, the output should be:
97 ['34', '67', '55', '33', '12', '98']
98 ('34', '67', '55', '33', '12', '98')
99
100 Hints:
101 In case of input data being supplied to the question, it should be assumed to be a console input.
102 tuple() method can convert list to tuple
103
104 Solution:
105 values=raw_input()
106 l=values.split(",")
107 t=tuple(l)
108 print l
109 print t
110 #-----#
111
112 #-----#

```

```

113 Question 5
114 Level 1
115
116 Question:
117 Define a class which has at least two methods:
118 getString: to get a string from console input
119 printString: to print the string in upper case.
120 Also please include simple test function to test the class methods.
121
122 Hints:
123 Use __init__ method to construct some parameters
124
125 Solution:
126 class InputOutString(object):
127     def __init__(self):
128         self.s = ""
129
130     def getString(self):
131         self.s = raw_input()
132
133     def printString(self):
134         print self.s.upper()
135
136 strObj = InputOutString()
137 strObj.getString()
138 strObj.printString()
139 #-----#
140
141 #-----#
142 Question 6
143 Level 2
144
145 Question:
146 Write a program that calculates and prints the value according to the given formula:
147  $Q = \text{Square root of } [(2 * C * D)/H]$ 
148 Following are the fixed values of C and H:
149 C is 50. H is 30.
150 D is the variable whose values should be input to your program in a comma-separated sequence.
151 Example
152 Let us assume the following comma separated input sequence is given to the program:
153 100,150,180
154 The output of the program should be:
155 18,22,24
156
157 Hints:
158 If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it s
159 In case of input data being supplied to the question, it should be assumed to be a console input.
160
161 Solution:
162 #!/usr/bin/env python
163 import math
164 c=50
165 h=30
166 value = []
167 items=[x for x in raw_input().split(',')]
168 for d in items:
169     value.append(str(int(round(math.sqrt(2*c*float(d)/h))))))
170
171 print ','.join(value)
172 #-----#
173
174 #-----#
175 Question 7
176 Level 2
177
178 Question:
179 Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column o

```

```

180 Note: i=0,1.., X-1; j=0,1,iY-1.
181 Example
182 Suppose the following inputs are given to the program:
183 3,5
184 Then, the output of the program should be:
185 [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]
186
187 Hints:
188 Note: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form.
189
190 Solution:
191 input_str = raw_input()
192 dimensions=[int(x) for x in input_str.split(',')]
193 rowNum=dimensions[0]
194 colNum=dimensions[1]
195 multilist = [[0 for col in range(colNum)] for row in range(rowNum)]
196
197 for row in range(rowNum):
198     for col in range(colNum):
199         multilist[row][col]= row*col
200
201 print multilist
202 #-----#
203
204 #-----#
205 Question 8
206 Level 2
207
208 Question:
209 Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting
210 Suppose the following input is supplied to the program:
211 without,hello,bag,world
212 Then, the output should be:
213 bag,hello,without,world
214
215 Hints:
216 In case of input data being supplied to the question, it should be assumed to be a console input.
217
218 Solution:
219 items=[x for x in raw_input().split(',')]
220 items.sort()
221 print ','.join(items)
222 #-----#
223
224 #-----#
225 Question 9
226 Level 2
227
228 QuestionE9
229 Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.
230 Suppose the following input is supplied to the program:
231 Hello world
232 Practice makes perfect
233 Then, the output should be:
234 HELLO WORLD
235 PRACTICE MAKES PERFECT
236
237 Hints:
238 In case of input data being supplied to the question, it should be assumed to be a console input.
239
240 Solution:
241 lines = []
242 while True:
243     s = raw_input()
244     if s:
245         lines.append(s.upper())
246     else:

```

```
247         break;
248
249 for sentence in lines:
250     print sentence
251 #-----#
252
253 #-----#
254 Question 10
255 Level 2
256
257 Question:
258 Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and
259 Suppose the following input is supplied to the program:
260 hello world and practice makes perfect and hello world again
261 Then, the output should be:
262 again and hello makes perfect practice world
263
264 Hints:
265 In case of input data being supplied to the question, it should be assumed to be a console input.
266 We use set container to remove duplicated data automatically and then use sorted() to sort the data.
267
268 Solution:
269 s = raw_input()
270 words = [word for word in s.split(" ")]
271 print " ".join(sorted(list(set(words))))
272 #-----#
273
274 #-----#
275 Question 11
276 Level 2
277
278 Question:
279 Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible b
280 Example:
281 0100,0011,1010,1001
282 Then the output should be:
283 1010
284 Notes: Assume the data is input by console.
285
286 Hints:
287 In case of input data being supplied to the question, it should be assumed to be a console input.
288
289 Solution:
290 value = []
291 items=[x for x in raw_input().split(',')]
292 for p in items:
293     intp = int(p, 2)
294     if not intp%5:
295         value.append(p)
296
297 print ','.join(value)
298 #-----#
299
300 #-----#
301 Question 12
302 Level 2
303
304 Question:
305 Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even numbe
306 The numbers obtained should be printed in a comma-separated sequence on a single line.
307
308 Hints:
309 In case of input data being supplied to the question, it should be assumed to be a console input.
310
311 Solution:
312 values = []
313 for i in range(1000, 3001):
```

```
314     s = str(i)
315     if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==0):
316         values.append(s)
317 print ",".join(values)
318 #-----#
319
320 #-----#
321 Question 13
322 Level 2
323
324 Question:
325 Write a program that accepts a sentence and calculate the number of letters and digits.
326 Suppose the following input is supplied to the program:
327 hello world! 123
328 Then, the output should be:
329 LETTERS 10
330 DIGITS 3
331
332 Hints:
333 In case of input data being supplied to the question, it should be assumed to be a console input.
334
335 Solution:
336 s = raw_input()
337 d={"DIGITS":0, "LETTERS":0}
338 for c in s:
339     if c.isdigit():
340         d["DIGITS"]+=1
341     elif c.isalpha():
342         d["LETTERS"]+=1
343     else:
344         pass
345 print "LETTERS", d["LETTERS"]
346 print "DIGITS", d["DIGITS"]
347 #-----#
348
349 #-----#
350 Question 14
351 Level 2
352
353 Question:
354 Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.
355 Suppose the following input is supplied to the program:
356 Hello world!
357 Then, the output should be:
358 UPPER CASE 1
359 LOWER CASE 9
360
361 Hints:
362 In case of input data being supplied to the question, it should be assumed to be a console input.
363
364 Solution:
365 s = raw_input()
366 d={"UPPER CASE":0, "LOWER CASE":0}
367 for c in s:
368     if c.isupper():
369         d["UPPER CASE"]+=1
370     elif c.islower():
371         d["LOWER CASE"]+=1
372     else:
373         pass
374 print "UPPER CASE", d["UPPER CASE"]
375 print "LOWER CASE", d["LOWER CASE"]
376 #-----#
377
378 #-----#
379 Question 15
380 Level 2
```

```
381
382 Question:
383 Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.
384 Suppose the following input is supplied to the program:
385 9
386 Then, the output should be:
387 11106
388
389 Hints:
390 In case of input data being supplied to the question, it should be assumed to be a console input.
391
392 Solution:
393 a = raw_input()
394 n1 = int( "%s" % a )
395 n2 = int( "%s%s" % (a,a) )
396 n3 = int( "%s%s%s" % (a,a,a) )
397 n4 = int( "%s%s%s%s" % (a,a,a,a) )
398 print n1+n2+n3+n4
399 #-----#
400
401 #-----#
402 Question 16
403 Level 2
404
405 Question:
406 Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.
407 Suppose the following input is supplied to the program:
408 1,2,3,4,5,6,7,8,9
409 Then, the output should be:
410 1,3,5,7,9
411
412 Hints:
413 In case of input data being supplied to the question, it should be assumed to be a console input.
414
415 Solution:
416 values = raw_input()
417 numbers = [x for x in values.split(",") if int(x)%2!=0]
418 print ",".join(numbers)
419 #-----#
420
421 Question 17
422 Level 2
423
424 Question:
425 Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is sh
426 D 100
427 W 200
428
429 D means deposit while W means withdrawal.
430 Suppose the following input is supplied to the program:
431 D 300
432 D 300
433 W 200
434 D 100
435 Then, the output should be:
436 500
437
438 Hints:
439 In case of input data being supplied to the question, it should be assumed to be a console input.
440
441 Solution:
442 netAmount = 0
443 while True:
444     s = raw_input()
445     if not s:
446         break
447     values = s.split(" ")
```

```

448     operation = values[0]
449     amount = int(values[1])
450     if operation=="D":
451         netAmount+=amount
452     elif operation=="W":
453         netAmount-=amount
454     else:
455         pass
456 print netAmount
457 #-----#
458
459 #-----#
460 Question 18
461 Level 3
462
463 Question:
464 A website requires the users to input username and password to register. Write a program to check the validity of password input by users.
465 Following are the criteria for checking the password:
466 1. At least 1 letter between [a-z]
467 2. At least 1 number between [0-9]
468 3. At least 1 letter between [A-Z]
469 4. At least 1 character from [!@#$%^&*]
470 5. Minimum length of transaction password: 6
471 6. Maximum length of transaction password: 12
472 Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match
473 the criteria will be accepted.
474 Example
475 If the following passwords are given as input to the program:
476 ABd1234@1,a F1#,2w3E*,2We3345
477 Then, the output of the program should be:
478 ABd1234@1
479
480 Hints:
481 In case of input data being supplied to the question, it should be assumed to be a console input.
482
483 Solutions:
484 import re
485 value = []
486 items=[x for x in raw_input().split(',')]
487 for p in items:
488     if len(p)<6 or len(p)>12:
489         continue
490     else:
491         pass
492     if not re.search("[a-z]",p):
493         continue
494     elif not re.search("[0-9]",p):
495         continue
496     elif not re.search("[A-Z]",p):
497         continue
498     elif not re.search("[!@#$%^&*]",p):
499         continue
500     elif re.search("\s",p):
501         continue
502     else:
503         pass
504     value.append(p)
505 print ",".join(value)
506 #-----#
507
508 #-----#
509 Question 19
510 Level 3
511
512 Question:
513 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.
514 1: Sort based on name;
515 2: Then sort based on age;

```



```

515 3: Then sort by score.
516 The priority is that name > age > score.
517 If the following tuples are given as input to the program:
518 Tom,19,80
519 John,20,90
520 Jony,17,91
521 Jony,17,93
522 Json,21,85
523 Then, the output of the program should be:
524 [('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]
525
526 Hints:
527 In case of input data being supplied to the question, it should be assumed to be a console input.
528 We use itemgetter to enable multiple sort keys.
529
530 Solutions:
531 from operator import itemgetter, attrgetter
532
533 l = []
534 while True:
535     s = raw_input()
536     if not s:
537         break
538     l.append(tuple(s.split(",")))
539
540 print sorted(l, key=itemgetter(0,1,2))
541 #-----#
542
543 #-----#
544 Question 20
545 Level 3
546
547 Question:
548 Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.
549
550 Hints:
551 Consider use yield
552
553 Solution:
554 def putNumbers(n):
555     i = 0
556     while i<n:
557         j=i
558         i=i+1
559         if j%7==0:
560             yield j
561
562 for i in reverse(100):
563     print i
564 #-----#
565
566 #-----#
567 Question 21
568 Level 3
569
570 Question 20
571 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
572 UP 5
573 DOWN 3
574 LEFT 3
575 RIGHT 2
576 i
577 The numbers after the direction are steps. Please write a program to compute the distance from current position after a sequence of movemen
578 Example:
579 If the following tuples are given as input to the program:
580 UP 5
581 DOWN 3

```

```

582 LEFT 3
583 RIGHT 2
584 Then, the output of the program should be:
585 2
586
587 Hints:
588 In case of input data being supplied to the question, it should be assumed to be a console input.
589
590 Solution:
591 import math
592 pos = [0,0]
593 while True:
594     s = raw_input()
595     if not s:
596         break
597     movement = s.split(" ")
598     direction = movement[0]
599     steps = int(movement[1])
600     if direction=="UP":
601         pos[0]+=steps
602     elif direction=="DOWN":
603         pos[0]-=steps
604     elif direction=="LEFT":
605         pos[1]-=steps
606     elif direction=="RIGHT":
607         pos[1]+=steps
608     else:
609         pass
610
611 print int(round(math.sqrt(pos[1]**2+pos[0]**2)))
612 #-----#
613
614 #-----#
615 Question 22
616 Level 3
617
618 Question:
619 Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.
620 Suppose the following input is supplied to the program:
621 New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.
622 Then, the output should be:
623 2:2
624 3.:1
625 3?:1
626 New:1
627 Python:5
628 Read:1
629 and:1
630 between:1
631 choosing:1
632 or:2
633 to:1
634
635 Hints
636 In case of input data being supplied to the question, it should be assumed to be a console input.
637
638 Solution:
639 freq = {} # frequency of words in text
640 line = raw_input()
641 for word in line.split():
642     freq[word] = freq.get(word,0)+1
643
644 words = freq.keys()
645 words.sort()
646
647 for w in words:
648     print "%s:%d" % (w,freq[w])

```

```
649 #-----#
650
651 #-----#
652 Question 23
653 level 1
654
655 Question:
656     Write a method which can calculate square value of number
657
658 Hints:
659     Using the ** operator
660
661 Solution:
662 def square(num):
663     return num ** 2
664
665 print square(2)
666 print square(3)
667 #-----#
668
669 #-----#
670 Question 24
671 Level 1
672
673 Question:
674     Python has many built-in functions, and if you do not know how to use it, you can read document online or find some books. But Python h
675     Please write a program to print some Python built-in functions documents, such as abs(), int(), raw_input()
676     And add document for your own function
677
678 Hints:
679     The built-in document method is __doc__
680
681 Solution:
682 print abs.__doc__
683 print int.__doc__
684 print raw_input.__doc__
685
686 def square(num):
687     '''Return the square value of the input number.
688
689     The input number must be integer.
690     '''
691     return num ** 2
692
693 print square(2)
694 print square.__doc__
695 #-----#
696
697 #-----#
698 Question 25
699 Level 1
700
701 Question:
702     Define a class, which have a class parameter and have a same instance parameter.
703
704 Hints:
705     Define a instance parameter, need add it in __init__ method
706     You can init a object with construct parameter or set the value later
707
708 Solution:
709 class Person:
710     # Define the class parameter "name"
711     name = "Person"
712
713     def __init__(self, name = None):
714         # self.name is the instance parameter
715         self.name = name
```

```
716
717 jeffrey = Person("Jeffrey")
718 print "%s name is %s" % (Person.name, jeffrey.name)
719
720 nico = Person()
721 nico.name = "Nico"
722 print "%s name is %s" % (Person.name, nico.name)
723 #-----#
724
725 #-----#
726 Question:
727 Define a function which can compute the sum of two numbers.
728
729 Hints:
730 Define a function with two numbers as arguments. You can compute the sum in the function and return the value.
731
732 Solution
733 def SumFunction(number1, number2):
734     return number1+number2
735
736 print SumFunction(1,2)
737
738 #-----#
739 Question:
740 Define a function that can convert a integer into a string and print it in console.
741
742 Hints:
743
744 Use str() to convert a number to string.
745
746 Solution
747 def printValue(n):
748     print str(n)
749
750 printValue(3)
751
752
753 #-----#
754 Question:
755 Define a function that can convert a integer into a string and print it in console.
756
757 Hints:
758
759 Use str() to convert a number to string.
760
761 Solution
762 def printValue(n):
763     print str(n)
764
765 printValue(3)
766
767 #-----#
768 2.10
769
770 Question:
771 Define a function that can receive two integral numbers in string form and compute their sum and then print it in console.
772
773 Hints:
774
775 Use int() to convert a string to integer.
776
777 Solution
778 def printValue(s1,s2):
779     print int(s1)+int(s2)
780
781 printValue("3","4") #7
782
```

```
783
784 #-----#
785 2.10
786
787
788 Question:
789 Define a function that can accept two strings as input and concatenate them and then print it in console.
790
791 Hints:
792
793 Use + to concatenate the strings
794
795 Solution
796 def printValue(s1,s2):
797     print s1+s2
798
799 printValue("3","4") #34
800
801 #-----#
802 2.10
803
804
805 Question:
806 Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same le
807
808 Hints:
809
810 Use len() function to get the length of a string
811
812 Solution
813 def printValue(s1,s2):
814     len1 = len(s1)
815     len2 = len(s2)
816     if len1>len2:
817         print s1
818     elif len2>len1:
819         print s2
820     else:
821         print s1
822         print s2
823
824
825 printValue("one","three")
826
827
828
829 #-----#
830 2.10
831
832 Question:
833 Define a function that can accept an integer number as input and print the "It is an even number" if the number is even, otherwise print "I
834
835 Hints:
836
837 Use % operator to check if a number is even or odd.
838
839 Solution
840 def checkValue(n):
841     if n%2 == 0:
842         print "It is an even number"
843     else:
844         print "It is an odd number"
845
846
847 checkValue(7)
848
849
```

```
850 #-----#
851 2.10
852
853 Question:
854 Define a function which can print a dictionary where the keys are numbers between 1 and 3 (both included) and the values are square of keys
855
856 Hints:
857
858 Use dict[key]=value pattern to put entry into a dictionary.
859 Use ** operator to get power of a number.
860
861 Solution
862 def printDict():
863     d=dict()
864     d[1]=1
865     d[2]=2**2
866     d[3]=3**2
867     print d
868
869
870 printDict()
871
872
873
874
875
876 #-----#
877 2.10
878
879 Question:
880 Define a function which can print a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of key
881
882 Hints:
883
884 Use dict[key]=value pattern to put entry into a dictionary.
885 Use ** operator to get power of a number.
886 Use range() for loops.
887
888 Solution
889 def printDict():
890     d=dict()
891     for i in range(1,21):
892         d[i]=i**2
893     print d
894
895
896 printDict()
897
898
899 #-----#
900 2.10
901
902 Question:
903 Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of
904
905 Hints:
906
907 Use dict[key]=value pattern to put entry into a dictionary.
908 Use ** operator to get power of a number.
909 Use range() for loops.
910 Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.
911
912 Solution
913 def printDict():
914     d=dict()
915     for i in range(1,21):
916         d[i]=i**2
```

```
917         for (k,v) in d.items():
918             print v
919
920
921 printDict()
922
923 #-----#
924 2.10
925
926 Question:
927 Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of
928
929 Hints:
930
931 Use dict[key]=value pattern to put entry into a dictionary.
932 Use ** operator to get power of a number.
933 Use range() for loops.
934 Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.
935
936 Solution
937 def printDict():
938     d=dict()
939     for i in range(1,21):
940         d[i]=i**2
941     for k in d.keys():
942         print k
943
944
945 printDict()
946
947
948 #-----#
949 2.10
950
951 Question:
952 Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included).
953
954 Hints:
955
956 Use ** operator to get power of a number.
957 Use range() for loops.
958 Use list.append() to add values into a list.
959
960 Solution
961 def printList():
962     li=list()
963     for i in range(1,21):
964         li.append(i**2)
965     print li
966
967
968 printList()
969
970 #-----#
971 2.10
972
973 Question:
974 Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function need
975
976 Hints:
977
978 Use ** operator to get power of a number.
979 Use range() for loops.
980 Use list.append() to add values into a list.
981 Use [n1:n2] to slice a list
982
983 Solution
```

```
984 def printList():
985     li=list()
986     for i in range(1,21):
987         li.append(i**2)
988     print li[:5]
989
990
991 printList()
992
993
994 #-----#
995 2.10
996
997 Question:
998 Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function need
999
1000 Hints:
1001
1002 Use ** operator to get power of a number.
1003 Use range() for loops.
1004 Use list.append() to add values into a list.
1005 Use [n1:n2] to slice a list
1006
1007 Solution
1008 def printList():
1009     li=list()
1010     for i in range(1,21):
1011         li.append(i**2)
1012     print li[-5:]
1013
1014
1015 printList()
1016
1017
1018 #-----#
1019 2.10
1020
1021 Question:
1022 Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function need
1023
1024 Hints:
1025
1026 Use ** operator to get power of a number.
1027 Use range() for loops.
1028 Use list.append() to add values into a list.
1029 Use [n1:n2] to slice a list
1030
1031 Solution
1032 def printList():
1033     li=list()
1034     for i in range(1,21):
1035         li.append(i**2)
1036     print li[5:]
1037
1038
1039 printList()
1040
1041
1042 #-----#
1043 2.10
1044
1045 Question:
1046 Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20 (both included).
1047
1048 Hints:
1049
1050 Use ** operator to get power of a number.
```



```
1051 Use range() for loops.
1052 Use list.append() to add values into a list.
1053 Use tuple() to get a tuple from a list.
1054
1055 Solution
1056 def printTuple():
1057     li=list()
1058     for i in range(1,21):
1059         li.append(i**2)
1060     print tuple(li)
1061
1062 printTuple()
1063
1064
1065
1066 #-----#
1067 2.10
1068
1069 Question:
1070 With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.
1071
1072 Hints:
1073
1074 Use [n1:n2] notation to get a slice from a tuple.
1075
1076 Solution
1077 tp=(1,2,3,4,5,6,7,8,9,10)
1078 tp1=tp[:5]
1079 tp2=tp[5:]
1080 print tp1
1081 print tp2
1082
1083
1084 #-----#
1085 2.10
1086
1087 Question:
1088 Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).
1089
1090 Hints:
1091
1092 Use "for" to iterate the tuple
1093 Use tuple() to generate a tuple from a list.
1094
1095 Solution
1096 tp=(1,2,3,4,5,6,7,8,9,10)
1097 li=list()
1098 for i in tp:
1099     if tp[i]%2==0:
1100         li.append(tp[i])
1101
1102 tp2=tuple(li)
1103 print tp2
1104
1105
1106
1107 #-----#
1108 2.14
1109
1110 Question:
1111 Write a program which accepts a string as input to print "Yes" if the string is "yes" or "YES" or "Yes", otherwise print "No".
1112
1113 Hints:
1114
1115 Use if statement to judge condition.
1116
1117 Solution
```

```
1118 s= raw_input()
1119 if s=="yes" or s=="YES" or s=="Yes":
1120     print "Yes"
1121 else:
1122     print "No"
1123
1124
1125
1126 #-----#
1127 3.4
1128
1129 Question:
1130 Write a program which can filter even numbers in a list by using filter function. The list is: [1,2,3,4,5,6,7,8,9,10].
1131
1132 Hints:
1133
1134 Use filter() to filter some elements in a list.
1135 Use lambda to define anonymous functions.
1136
1137 Solution
1138 li = [1,2,3,4,5,6,7,8,9,10]
1139 evenNumbers = filter(lambda x: x%2==0, li)
1140 print evenNumbers
1141
1142
1143 #-----#
1144 3.4
1145
1146 Question:
1147 Write a program which can map() to make a list whose elements are square of elements in [1,2,3,4,5,6,7,8,9,10].
1148
1149 Hints:
1150
1151 Use map() to generate a list.
1152 Use lambda to define anonymous functions.
1153
1154 Solution
1155 li = [1,2,3,4,5,6,7,8,9,10]
1156 squaredNumbers = map(lambda x: x**2, li)
1157 print squaredNumbers
1158
1159 #-----#
1160 3.5
1161
1162 Question:
1163 Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10].
1164
1165 Hints:
1166
1167 Use map() to generate a list.
1168 Use filter() to filter elements of a list.
1169 Use lambda to define anonymous functions.
1170
1171 Solution
1172 li = [1,2,3,4,5,6,7,8,9,10]
1173 evenNumbers = map(lambda x: x**2, filter(lambda x: x%2==0, li))
1174 print evenNumbers
1175
1176
1177
1178 #-----#
1179 3.5
1180
1181
1182 Question:
1183 Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included).
1184
```

```
1185 Hints:
1186
1187 Use filter() to filter elements of a list.
1188 Use lambda to define anonymous functions.
1189
1190 Solution
1191 evenNumbers = filter(lambda x: x%2==0, range(1,21))
1192 print evenNumbers
1193
1194
1195 #-----#
1196 3.5
1197
1198 Question:
1199 Write a program which can map() to make a list whose elements are square of numbers between 1 and 20 (both included).
1200
1201 Hints:
1202
1203 Use map() to generate a list.
1204 Use lambda to define anonymous functions.
1205
1206 Solution
1207 squaredNumbers = map(lambda x: x**2, range(1,21))
1208 print squaredNumbers
1209
1210
1211
1212
1213 #-----#
1214 7.2
1215
1216 Question:
1217 Define a class named American which has a static method called printNationality.
1218
1219 Hints:
1220
1221 Use @staticmethod decorator to define class static method.
1222
1223 Solution
1224 class American(object):
1225     @staticmethod
1226     def printNationality():
1227         print "America"
1228
1229 anAmerican = American()
1230 anAmerican.printNationality()
1231 American.printNationality()
1232
1233
1234
1235
1236 #-----#
1237
1238 7.2
1239
1240 Question:
1241 Define a class named American and its subclass NewYorker.
1242
1243 Hints:
1244
1245 Use class Subclass(ParentClass) to define a subclass.
1246
1247 Solution:
1248
1249 class American(object):
1250     pass
1251
```

```
1252 class NewYorker(American):
1253     pass
1254
1255 anAmerican = American()
1256 aNewYorker = NewYorker()
1257 print anAmerican
1258 print aNewYorker
1259
1260
1261
1262
1263 #-----#
1264
1265
1266 7.2
1267
1268 Question:
1269 Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.
1270
1271 Hints:
1272
1273 Use def methodName(self) to define a method.
1274
1275 Solution:
1276
1277 class Circle(object):
1278     def __init__(self, r):
1279         self.radius = r
1280
1281     def area(self):
1282         return self.radius**2*3.14
1283
1284 aCircle = Circle(2)
1285 print aCircle.area()
1286
1287
1288
1289
1290
1291
1292 #-----#
1293
1294 7.2
1295
1296 Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.
1297
1298 Hints:
1299
1300 Use def methodName(self) to define a method.
1301
1302 Solution:
1303
1304 class Rectangle(object):
1305     def __init__(self, l, w):
1306         self.length = l
1307         self.width = w
1308
1309     def area(self):
1310         return self.length*self.width
1311
1312 aRectangle = Rectangle(2,10)
1313 print aRectangle.area()
1314
1315
1316
1317
1318 #-----#
```

```
1319
1320 7.2
1321
1322 Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes ha
1323
1324 Hints:
1325
1326 To override a method in super class, we can define a method with the same name in the super class.
1327
1328 Solution:
1329
1330 class Shape(object):
1331     def __init__(self):
1332         pass
1333
1334     def area(self):
1335         return 0
1336
1337 class Square(Shape):
1338     def __init__(self, l):
1339         Shape.__init__(self)
1340         self.length = l
1341
1342     def area(self):
1343         return self.length*self.length
1344
1345 aSquare= Square(3)
1346 print aSquare.area()
1347
1348
1349
1350
1351
1352
1353
1354
1355 #-----#
1356
1357
1358 Please raise a RuntimeError exception.
1359
1360 Hints:
1361
1362 Use raise() to raise an exception.
1363
1364 Solution:
1365
1366 raise RuntimeError('something wrong')
1367
1368
1369
1370 #-----#
1371 Write a function to compute 5/0 and use try/except to catch the exceptions.
1372
1373 Hints:
1374
1375 Use try/except to catch exceptions.
1376
1377 Solution:
1378
1379 def throws():
1380     return 5/0
1381
1382 try:
1383     throws()
1384 except ZeroDivisionError:
1385     print "division by zero!"
```

```
1386 except Exception, err:
1387     print 'Caught an exception'
1388 finally:
1389     print 'In finally block for cleanup'
1390
1391
1392 #-----#
1393 Define a custom exception class which takes a string message as attribute.
1394
1395 Hints:
1396
1397 To define a custom exception, we need to define a class inherited from Exception.
1398
1399 Solution:
1400
1401 class MyError(Exception):
1402     """My own exception class
1403
1404     Attributes:
1405         msg -- explanation of the error
1406     """
1407
1408     def __init__(self, msg):
1409         self.msg = msg
1410
1411 error = MyError("something wrong")
1412
1413 #-----#
1414 Question:
1415
1416 Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the user name of a given
1417
1418 Example:
1419 If the following email address is given as input to the program:
1420
1421 john@google.com
1422
1423 Then, the output of the program should be:
1424
1425 john
1426
1427 In case of input data being supplied to the question, it should be assumed to be a console input.
1428
1429 Hints:
1430
1431 Use \w to match letters.
1432
1433 Solution:
1434 import re
1435 emailAddress = raw_input()
1436 pat2 = "(\w+)@((\w+\.)+(com))"
1437 r2 = re.match(pat2,emailAddress)
1438 print r2.group(1)
1439
1440
1441 #-----#
1442 Question:
1443
1444 Assuming that we have some email addresses in the "username@companyname.com" format, please write program to print the company name of a gi
1445
1446 Example:
1447 If the following email address is given as input to the program:
1448
1449 john@google.com
1450
1451 Then, the output of the program should be:
1452
```

```
1453 google
1454
1455 In case of input data being supplied to the question, it should be assumed to be a console input.
1456
1457 Hints:
1458
1459 Use \w to match letters.
1460
1461 Solution:
1462 import re
1463 emailAddress = raw_input()
1464 pat2 = "(\w+)@(\w+)\.(com)"
1465 r2 = re.match(pat2,emailAddress)
1466 print r2.group(2)
1467
1468
1469
1470
1471 #-----#
1472 Question:
1473
1474 Write a program which accepts a sequence of words separated by whitespace as input to print the words composed of digits only.
1475
1476 Example:
1477 If the following words is given as input to the program:
1478
1479 2 cats and 3 dogs.
1480
1481 Then, the output of the program should be:
1482
1483 ['2', '3']
1484
1485 In case of input data being supplied to the question, it should be assumed to be a console input.
1486
1487 Hints:
1488
1489 Use re.findall() to find all substring using regex.
1490
1491 Solution:
1492 import re
1493 s = raw_input()
1494 print re.findall("\d+",s)
1495
1496
1497 #-----#
1498 Question:
1499
1500 Print a unicode string "hello world".
1501
1502 Hints:
1503
1504 Use u'strings' format to define unicode string.
1505
1506 Solution:
1507
1508 unicodeString = u"hello world!"
1509 print unicodeString
1510
1511 #-----#
1512 Write a program to read an ASCII string and to convert it to a unicode string encoded by utf-8.
1513
1514 Hints:
1515
1516 Use unicode() function to convert.
1517
1518 Solution:
```

```
1520
1521 s = raw_input()
1522 u = unicode( s ,"utf-8")
1523 print u
1524
1525 #-----#
1526 Question:
1527
1528 Write a special comment to indicate a Python source code file is in unicode.
1529
1530 Hints:
1531
1532 Solution:
1533
1534 # -*- coding: utf-8 -*-
1535
1536 #-----#
1537 Question:
1538
1539 Write a program to compute  $1/2+2/3+3/4+\dots+n/n+1$  with a given n input by console ( $n>0$ ).
1540
1541 Example:
1542 If the following n is given as input to the program:
1543
1544 5
1545
1546 Then, the output of the program should be:
1547
1548 3.55
1549
1550 In case of input data being supplied to the question, it should be assumed to be a console input.
1551
1552 Hints:
1553 Use float() to convert an integer to a float
1554
1555 Solution:
1556
1557 n=int(raw_input())
1558 sum=0.0
1559 for i in range(1,n+1):
1560     sum += float(float(i)/(i+1))
1561 print sum
1562
1563
1564 #-----#
1565 Question:
1566
1567 Write a program to compute:
1568
1569  $f(n)=f(n-1)+100$  when  $n>0$ 
1570 and  $f(0)=1$ 
1571
1572 with a given n input by console ( $n>0$ ).
1573
1574 Example:
1575 If the following n is given as input to the program:
1576
1577 5
1578
1579 Then, the output of the program should be:
1580
1581 500
1582
1583 In case of input data being supplied to the question, it should be assumed to be a console input.
1584
1585 Hints:
1586 We can define recursive function in Python.
```



```
1587
1588 Solution:
1589
1590 def f(n):
1591     if n==0:
1592         return 0
1593     else:
1594         return f(n-1)+100
1595
1596 n=int(raw_input())
1597 print f(n)
1598
1599 #-----#
1600
1601 Question:
1602
1603
1604 The Fibonacci Sequence is computed based on the following formula:
1605
1606
1607  $f(n)=0$  if  $n=0$ 
1608  $f(n)=1$  if  $n=1$ 
1609  $f(n)=f(n-1)+f(n-2)$  if  $n>1$ 
1610
1611 Please write a program to compute the value of  $f(n)$  with a given  $n$  input by console.
1612
1613 Example:
1614 If the following  $n$  is given as input to the program:
1615
1616 7
1617
1618 Then, the output of the program should be:
1619
1620 13
1621
1622 In case of input data being supplied to the question, it should be assumed to be a console input.
1623
1624 Hints:
1625 We can define recursive function in Python.
1626
1627
1628 Solution:
1629
1630 def f(n):
1631     if n == 0: return 0
1632     elif n == 1: return 1
1633     else: return f(n-1)+f(n-2)
1634
1635 n=int(raw_input())
1636 print f(n)
1637
1638
1639 #-----#
1640
1641 #-----#
1642
1643 Question:
1644
1645 The Fibonacci Sequence is computed based on the following formula:
1646
1647
1648  $f(n)=0$  if  $n=0$ 
1649  $f(n)=1$  if  $n=1$ 
1650  $f(n)=f(n-1)+f(n-2)$  if  $n>1$ 
1651
1652 Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given  $n$  input by console.
1653
```

```
1654 Example:
1655 If the following n is given as input to the program:
1656
1657 7
1658
1659 Then, the output of the program should be:
1660
1661 0,1,1,2,3,5,8,13
1662
1663
1664 Hints:
1665 We can define recursive function in Python.
1666 Use list comprehension to generate a list from an existing list.
1667 Use string.join() to join a list of strings.
1668
1669 In case of input data being supplied to the question, it should be assumed to be a console input.
1670
1671 Solution:
1672
1673 def f(n):
1674     if n == 0: return 0
1675     elif n == 1: return 1
1676     else: return f(n-1)+f(n-2)
1677
1678 n=int(raw_input())
1679 values = [str(f(x)) for x in range(0, n+1)]
1680 print ",".join(values)
1681
1682
1683 #-----#
1684
1685 Question:
1686
1687 Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.
1688
1689 Example:
1690 If the following n is given as input to the program:
1691
1692 10
1693
1694 Then, the output of the program should be:
1695
1696 0,2,4,6,8,10
1697
1698 Hints:
1699 Use yield to produce the next value in generator.
1700
1701 In case of input data being supplied to the question, it should be assumed to be a console input.
1702
1703 Solution:
1704
1705 def EvenGenerator(n):
1706     i=0
1707     while i<=n:
1708         if i%2==0:
1709             yield i
1710             i+=1
1711
1712
1713 n=int(raw_input())
1714 values = []
1715 for i in EvenGenerator(n):
1716     values.append(str(i))
1717
1718 print ",".join(values)
1719
1720
```

```
1721 #-----#
1722
1723 Question:
1724
1725 Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while
1726
1727 Example:
1728 If the following n is given as input to the program:
1729
1730 100
1731
1732 Then, the output of the program should be:
1733
1734 0,35,70
1735
1736 Hints:
1737 Use yield to produce the next value in generator.
1738
1739 In case of input data being supplied to the question, it should be assumed to be a console input.
1740
1741 Solution:
1742
1743 def NumGenerator(n):
1744     for i in range(n+1):
1745         if i%5==0 and i%7==0:
1746             yield i
1747
1748 n=int(raw_input())
1749 values = []
1750 for i in NumGenerator(n):
1751     values.append(str(i))
1752
1753 print ",".join(values)
1754
1755 #-----#
1756
1757 Question:
1758
1759
1760
1761 Please write assert statements to verify that every number in the list [2,4,6,8] is even.
1762
1763
1764
1765 Hints:
1766 Use "assert expression" to make assertion.
1767
1768
1769 Solution:
1770
1771 li = [2,4,6,8]
1772 for i in li:
1773     assert i%2==0
1774
1775 #-----#
1776
1777 Question:
1778
1779 Please write a program which accepts basic mathematic expression from console and print the evaluation result.
1780
1781 Example:
1782 If the following string is given as input to the program:
1783
1784 35+3
1785
1786 Then, the output of the program should be:
1787
```

```
1788 38
1789
1790 Hints:
1791 Use eval() to evaluate an expression.
1792
1793
1794 Solution:
1795
1796 expression = raw_input()
1797 print eval(expression)
1798
1799
1800 #-----#
1801 Question:
1802
1803 Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be search
1804
1805
1806 Hints:
1807 Use if/elif to deal with conditions.
1808
1809
1810 Solution:
1811
1812 import math
1813 def bin_search(li, element):
1814     bottom = 0
1815     top = len(li)-1
1816     index = -1
1817     while top>=bottom and index==-1:
1818         mid = int(math.floor((top+bottom)/2.0))
1819         if li[mid]==element:
1820             index = mid
1821         elif li[mid]>element:
1822             top = mid-1
1823         else:
1824             bottom = mid+1
1825
1826     return index
1827
1828 li=[2,5,7,9,11,17,222]
1829 print bin_search(li,11)
1830 print bin_search(li,12)
1831
1832
1833
1834
1835 #-----#
1836 Question:
1837
1838 Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be search
1839
1840
1841 Hints:
1842 Use if/elif to deal with conditions.
1843
1844
1845 Solution:
1846
1847 import math
1848 def bin_search(li, element):
1849     bottom = 0
1850     top = len(li)-1
1851     index = -1
1852     while top>=bottom and index==-1:
1853         mid = int(math.floor((top+bottom)/2.0))
1854         if li[mid]==element:
```

```
1855         index = mid
1856     elif li[mid]>element:
1857         top = mid-1
1858     else:
1859         bottom = mid+1
1860
1861     return index
1862
1863 li=[2,5,7,9,11,17,222]
1864 print bin_search(li,11)
1865 print bin_search(li,12)
1866
1867
1868
1869
1870 #-----#
1871 Question:
1872
1873 Please generate a random float where the value is between 10 and 100 using Python math module.
1874
1875
1876
1877 Hints:
1878 Use random.random() to generate a random float in [0,1].
1879
1880
1881 Solution:
1882
1883 import random
1884 print random.random()*100
1885
1886 #-----#
1887 Question:
1888
1889 Please generate a random float where the value is between 5 and 95 using Python math module.
1890
1891
1892
1893 Hints:
1894 Use random.random() to generate a random float in [0,1].
1895
1896
1897 Solution:
1898
1899 import random
1900 print random.random()*100-5
1901
1902
1903 #-----#
1904 Question:
1905
1906 Please write a program to output a random even number between 0 and 10 inclusive using random module and list comprehension.
1907
1908
1909
1910 Hints:
1911 Use random.choice() to a random element from a list.
1912
1913
1914 Solution:
1915
1916 import random
1917 print random.choice([i for i in range(11) if i%2==0])
1918
1919
1920 #-----#
1921 Question:
```

```
1922
1923 Please write a program to output a random number, which is divisible by 5 and 7, between 0 and 10 inclusive using random module and list co
1924
1925
1926
1927 Hints:
1928 Use random.choice() to a random element from a list.
1929
1930
1931 Solution:
1932
1933 import random
1934 print random.choice([i for i in range(201) if i%5==0 and i%7==0])
1935
1936
1937
1938 #-----#
1939
1940 Question:
1941
1942 Please write a program to generate a list with 5 random numbers between 100 and 200 inclusive.
1943
1944
1945
1946 Hints:
1947 Use random.sample() to generate a list of random values.
1948
1949
1950 Solution:
1951
1952 import random
1953 print random.sample(range(100), 5)
1954
1955 #-----#
1956 Question:
1957
1958 Please write a program to randomly generate a list with 5 even numbers between 100 and 200 inclusive.
1959
1960
1961
1962 Hints:
1963 Use random.sample() to generate a list of random values.
1964
1965
1966 Solution:
1967
1968 import random
1969 print random.sample([i for i in range(100,201) if i%2==0], 5)
1970
1971
1972 #-----#
1973 Question:
1974
1975 Please write a program to randomly generate a list with 5 numbers, which are divisible by 5 and 7 , between 1 and 1000 inclusive.
1976
1977
1978
1979 Hints:
1980 Use random.sample() to generate a list of random values.
1981
1982
1983 Solution:
1984
1985 import random
1986 print random.sample([i for i in range(1,1001) if i%5==0 and i%7==0], 5)
1987
1988 #-----#
```

```
1989
1990 Question:
1991
1992 Please write a program to randomly print a integer number between 7 and 15 inclusive.
1993
1994
1995
1996 Hints:
1997 Use random.randrange() to a random integer in a given range.
1998
1999
2000 Solution:
2001
2002 import random
2003 print random.randrange(7,16)
2004
2005 #-----#
2006
2007 Question:
2008
2009 Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".
2010
2011
2012
2013 Hints:
2014 Use zlib.compress() and zlib.decompress() to compress and decompress a string.
2015
2016
2017 Solution:
2018
2019 import zlib
2020 s = 'hello world!hello world!hello world!hello world!'
2021 t = zlib.compress(s)
2022 print t
2023 print zlib.decompress(t)
2024
2025 #-----#
2026 Question:
2027
2028 Please write a program to print the running time of execution of "1+1" for 100 times.
2029
2030
2031
2032 Hints:
2033 Use timeit() function to measure the running time.
2034
2035 Solution:
2036
2037 from timeit import Timer
2038 t = Timer("for i in range(100):1+1")
2039 print t.timeit()
2040
2041 #-----#
2042 Question:
2043
2044 Please write a program to shuffle and print the list [3,6,7,8].
2045
2046
2047
2048 Hints:
2049 Use shuffle() function to shuffle a list.
2050
2051 Solution:
2052
2053 from random import shuffle
2054 li = [3,6,7,8]
2055 shuffle(li)
```

```
2056 print li
2057
2058 #-----#
2059 Question:
2060
2061 Please write a program to shuffle and print the list [3,6,7,8].
2062
2063
2064
2065 Hints:
2066 Use shuffle() function to shuffle a list.
2067
2068 Solution:
2069
2070 from random import shuffle
2071 li = [3,6,7,8]
2072 shuffle(li)
2073 print li
2074
2075
2076
2077 #-----#
2078 Question:
2079
2080 Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Ho
2081
2082 Hints:
2083 Use list[index] notation to get a element from a list.
2084
2085 Solution:
2086
2087 subjects=["I", "You"]
2088 verbs=["Play", "Love"]
2089 objects=["Hockey","Football"]
2090 for i in range(len(subjects)):
2091     for j in range(len(verbs)):
2092         for k in range(len(objects)):
2093             sentence = "%s %s %s." % (subjects[i], verbs[j], objects[k])
2094             print sentence
2095
2096
2097 #-----#
2098 Please write a program to print the list after removing delete even numbers in [5,6,77,45,22,12,24].
2099
2100 Hints:
2101 Use list comprehension to delete a bunch of element from a list.
2102
2103 Solution:
2104
2105 li = [5,6,77,45,22,12,24]
2106 li = [x for x in li if x%2!=0]
2107 print li
2108
2109 #-----#
2110 Question:
2111
2112 By using list comprehension, please write a program to print the list after removing delete numbers which are divisible by 5 and 7 in [12,2
2113
2114 Hints:
2115 Use list comprehension to delete a bunch of element from a list.
2116
2117 Solution:
2118
2119 li = [12,24,35,70,88,120,155]
2120 li = [x for x in li if x%5!=0 and x%7!=0]
2121 print li
2122
```



```
2123
2124 #-----#
2125 Question:
2126
2127 By using list comprehension, please write a program to print the list after removing the 0th, 2nd, 4th,6th numbers in [12,24,35,70,88,120,1
2128
2129 Hints:
2130 Use list comprehension to delete a bunch of element from a list.
2131 Use enumerate() to get (index, value) tuple.
2132
2133 Solution:
2134
2135 li = [12,24,35,70,88,120,155]
2136 li = [x for (i,x) in enumerate(li) if i%2!=0]
2137 print li
2138
2139 #-----#
2140
2141 Question:
2142
2143 By using list comprehension, please write a program generate a 3*5*8 3D array whose each element is 0.
2144
2145 Hints:
2146 Use list comprehension to make an array.
2147
2148 Solution:
2149
2150 array = [[ [0 for col in range(8)] for col in range(5)] for row in range(3)]
2151 print array
2152
2153 #-----#
2154 Question:
2155
2156 By using list comprehension, please write a program to print the list after removing the 0th,4th,5th numbers in [12,24,35,70,88,120,155].
2157
2158 Hints:
2159 Use list comprehension to delete a bunch of element from a list.
2160 Use enumerate() to get (index, value) tuple.
2161
2162 Solution:
2163
2164 li = [12,24,35,70,88,120,155]
2165 li = [x for (i,x) in enumerate(li) if i not in (0,4,5)]
2166 print li
2167
2168
2169
2170 #-----#
2171
2172 Question:
2173
2174 By using list comprehension, please write a program to print the list after removing the value 24 in [12,24,35,24,88,120,155].
2175
2176 Hints:
2177 Use list's remove method to delete a value.
2178
2179 Solution:
2180
2181 li = [12,24,35,24,88,120,155]
2182 li = [x for x in li if x!=24]
2183 print li
2184
2185
2186 #-----#
2187 Question:
2188
2189 With two given lists [1,3,6,78,35,55] and [12,24,35,24,88,120,155], write a program to make a list whose elements are intersection of the a
```

```
2190
2191 Hints:
2192 Use set() and "&=" to do set intersection operation.
2193
2194 Solution:
2195
2196 set1=set([1,3,6,78,35,55])
2197 set2=set([12,24,35,24,88,120,155])
2198 set1 &= set2
2199 li=list(set1)
2200 print li
2201
2202 #-----#
2203
2204 With a given list [12,24,35,24,88,120,155,88,120,155], write a program to print this list after removing all duplicate values with original
2205
2206 Hints:
2207 Use set() to store a number of values without duplicate.
2208
2209 Solution:
2210
2211 def removeDuplicate( li ):
2212     newli=[]
2213     seen = set()
2214     for item in li:
2215         if item not in seen:
2216             seen.add( item )
2217             newli.append(item)
2218
2219     return newli
2220
2221 li=[12,24,35,24,88,120,155,88,120,155]
2222 print removeDuplicate(li)
2223
2224 #-----#
2225
2226 Question:
2227
2228 Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male cla
2229
2230 Hints:
2231 Use Subclass(Parentclass) to define a child class.
2232
2233 Solution:
2234
2235 class Person(object):
2236     def getGender( self ):
2237         return "Unknown"
2238
2239 class Male( Person ):
2240     def getGender( self ):
2241         return "Male"
2242
2243 class Female( Person ):
2244     def getGender( self ):
2245         return "Female"
2246
2247 aMale = Male()
2248 aFemale= Female()
2249 print aMale.getGender()
2250 print aFemale.getGender()
2251
2252
2253 #-----#
2254
2255 Question:
2256
```

2257 Please write a program which count and print the numbers of each character in a string input by console.

2258

2259 Example:

2260 If the following string is given as input to the program:

2261

2262 abcdefgabc

2263

2264 Then, the output of the program should be:

2265

2266 a,2

2267 c,2

2268 b,2

2269 e,1

2270 d,1

2271 g,1

2272 f,1

2273

2274 Hints:

2275 Use dict to store key/value pairs.

2276 Use dict.get() method to lookup a key with default value.

2277

2278 Solution:

2279

```
2280 dic = {}
```

```
2281 s=raw_input()
```

```
2282 for s in s:
```

```
2283     dic[s] = dic.get(s,0)+1
```

```
2284 print '\n'.join(['%s,%s' % (k, v) for k, v in dic.items()])
```

2285

2286 #-----#

2287

2288 Question:

2289

2290 Please write a program which accepts a string from console and print it in reverse order.

2291

2292 Example:

2293 If the following string is given as input to the program:

2294

2295 rise to vote sir

2296

2297 Then, the output of the program should be:

2298

2299 ris etov ot esir

2300

2301 Hints:

2302 Use list[::-1] to iterate a list in a reverse order.

2303

2304 Solution:

2305

```
2306 s=raw_input()
```

```
2307 s = s[::-1]
```

```
2308 print s
```

2309

2310 #-----#

2311

2312 Question:

2313

2314 Please write a program which accepts a string from console and print the characters that have even indexes.

2315

2316 Example:

2317 If the following string is given as input to the program:

2318

2319 H1e2l3l4o5w6o7r8l9d

2320

2321 Then, the output of the program should be:

2322

2323 Helloworld

```
2324
2325 Hints:
2326 Use list[::-2] to iterate a list by step 2.
2327
2328 Solution:
2329
2330 s=raw_input()
2331 s = s[::-2]
2332 print s
2333 #-----#
2334
2335
2336 Question:
2337
2338 Please write a program which prints all permutations of [1,2,3]
2339
2340
2341 Hints:
2342 Use itertools.permutations() to get permutations of list.
2343
2344 Solution:
2345
2346 import itertools
2347 print list(itertools.permutations([1,2,3]))
2348
2349 #-----#
2350 Question:
2351
2352 Write a program to solve a classic ancient Chinese puzzle:
2353 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?
2354
2355 Hint:
2356 Use for loop to iterate all possible solutions.
2357
2358 Solution:
2359
2360 def solve(numheads,numlegs):
2361     ns='No solutions!'
2362     for i in range(numheads+1):
2363         j=numheads-i
2364         if 2*i+4*j==numlegs:
2365             return i,j
2366     return ns,ns
2367
2368 numheads=35
2369 numlegs=94
2370 solutions=solve(numheads,numlegs)
2371 print solutions
2372
2373 #-----#
2374
2375
```