X

(https://swayam.gov.in) (https://swayam.gov.in/nc_details/NPTEL)

gauravsharma727545@gmail.com

NPTEL (https://swayam.gov.in/explorer?ncCode=NPTEL) » Programming, Data Structures And Algorithms Using Python (course)

Announcements (announcements)

About the Course (https://swayam.gov.in/nd1_noc19_cs40/preview) Ask a Question (forum)

Progress (student/home) Mentor (student/mentor)

Course outline

- **How to access the portal**
- **Week 1: Introduction**
- **Week 1 Quiz**
- **Week 2: Basics of Python**
- **Week 2 Quiz**
- **Week 2 Programming Assignment**
- **Week 3: Lists, inductive function**

Online Test 2, Question 4

Due on 2019-09-26, 22:00 IST

definitions, sorting

Week 3 Programming Assignment

Week 4: Sorting, Tuples, Dictionaries, Passing Functions, List Comprehension

Week 4 Quiz

Week 4 Programming Assignment

Week 5: Exception handling, input/output, file handling, string processing

Week 5 Programming Assignment

Week 6: Backtracking, scope, data structures; stacks, queues and heaps

Week 6 Quiz

Week 7: Classes, objects and user defined datatypes

Week 7 Quiz

Instructions

This is the second of two online programming tests.

- These tests account for 25% of the total evaluation for the course.
- The duration of the test is 2 hours.
- The first test was from 9:30-11:30 am and the second is from 8:00-10:00 pm, on Thursday, 26 September 2019.
- You can attempt either of the tests. The best score will be counted..

Question 4

Recall that the positions in a list of length n are 0,1,...,n-1. We want to write a function `oddpositions(l)` that returns the elements at the odd positions in `l`. In other words, the function should return the list `[l[1],l[3],...]`. For instance `oddpositions([]) == []`, `oddpositions([7]) == []`, `oddpositions([8,11,8]) == [11]` and `oddpositions([19,3,44,44,3,19]) == [3,44,19]`. A recursive definition of `oddpositions` is given below. You have to fill in the missing argument for the recursive call.

```
def oddpositions(l):
  if len(l) <= 1:
    return([])
  else:
    return(...)
```

Open up the code submission box below and fill in the missing argument for the recursive call.

Private Test cases used for evaluation	**Input**	**Expected Output**	**Actual Output**	**Status**
Test Case 1	`oddpositions([13,42,12,16,18])`	`[42, 16]\n`	`[42, 16]\n`	Passed
Test Case 2	`oddpositions([1,2,3,4,5,6,7,8,9,10])`	`[2, 4, 6, 8, 10]\n`	`[2, 4, 6, 8, 10]\n`	Passed
Test Case 3	`oddpositions([2,3,4,5,6,7,8,9,10,11])`	`[3, 5, 7, 9, 11]\n`	`[3, 5, 7, 9, 11]\n`	Passed

Week 8: Dynamic programming, wrap-up

Week 8 Programming Assignment

Download videos

Text Transcripts

Online Programming Test - Sample

Online Programming Test 1, 26 Sep 2019, 09:30-11:30

Online Programming Test 2, 26 Sep 2019, 20:00-22:00

- Online Test 2, Question 1 (/noc19_cs40/progassignment?name=121)
- Online Test 2, Question 2 (/noc19_cs40/progassignment?name=122)
- Online Test 2, Question 3 (/noc19_cs40/progassignment?name=123)
- **Online Test 2, Question 4 (/noc19_cs40/progassignment?name=124)**
- Online Test 2, Question 5

Test Case 4	`oddpositions ([3,4,5,6])`	`[4, 6] \n`	`[4, 6] \n`	Passed

Due Date Exceeded.
4 out of 4 tests passed.
You scored 100.0/100.

Your last recorded submission was :

```
def oddpositions(l):
  if len(l) <= 1:
    return([])
  else:
    return(
        # Complete the recursive call below this line
l[1::2]
        # Complete the recursive call above this line
    )

import ast

def tolist(inp):
  inp = ast.literal_eval(inp)
  return(inp)

fncall = input()
lparen = fncall.find("(")
rparen = fncall.rfind(")")
fname = fncall[:lparen]
farg = fncall[lparen+1:rparen]

if fname == "oddpositions":
  arg = tolist(farg)
  print(oddpositions(arg))

```

(/noc19_cs40/progassignment?name=125)

- Online Test 2, Question 6 (/noc19_cs40/progassignment?name=126)
- Online Test 2, Question 7 (/noc19_cs40/progassignment?name=127)
- Online Test 2, Question 8 (/noc19_cs40/progassignment?name=128)
- Online Programming Test 2, 26 Sep 2019, 20:00-22:00 (unit?unit=111&lesson=129)