Register for Certification exam (https://examform.nptel.ac.in/2022_04/exam_form/dashboard)

Course outline

How does an NPTEL online course work? ()

Week 1 : Introduction ()

Week 1 Quiz ()

Week 2: Basics of Python ()

Week 2 Quiz ()

Week 2 Programming Assignment ()

Week 2 Programming Assignment (/noc22_cs70/progassignment?name=124)

Week 3: Lists, inductive function definitions, sorting ()

Week 2 Programming Assignment

Due on 2022-08-11, 23:59 IST

Write three Python functions as specified below. Paste the text for all three functions together into the submission window. Your function will be called automatically with various inputs and should return values as specified. Do not write commands to read any input or print any output.

- You may define additional auxiliary functions as needed.
- In all cases you may assume that the value passed to the function is of the expected type, so your function does not have to check for malformed inputs.
- For each function, there are normally some public test cases and some (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases. There are 10 private test cases, with equal weightage. You will get feedback about which private test cases pass or fail, though you cannot see the actual test cases.
- Ignore warnings about "Presentation errors".

1. Write a function `intreverse(n)` that takes as input a positive integer `n` and returns the integer obtained by reversing the digits in `n`.

   Here are some examples of how your function should work.

```
>>> intreverse(783)
387
>>> intreverse(242789)
987242
>>> intreverse(3)
3
```

2. Write a function `matched(s)` that takes as input a string `s` and checks if the brackets `"("` and `")"` in `s` are matched: that is, every `"("` has a matching `")"` after it and every `")"` has a matching `"("` before it. Your function should ignore all other symbols that appear in `s`. Your function should return `True` if `s` has mat brackets and `False` if it does not.

   Here are some examples to show how your function should work

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 ()

Week 8: Dynamic programming, wrap-up ()

Week 8 Programming Assignment ()

Here are some examples to show how your function should work.

```
>>> matched("zb%78")
True
>>> matched("(7)(a")
False
>>> matched("a)*(?")
False
>>> matched("((jkl)78(A)&l(8(dd(FJI:),):)?)")
True
```

3. Write a function `sumprimes(l)` that takes as input a list of integers `l` and retuns the sum of all the prime numbers in `l`.

   Here are some examples to show how your function should work.

```
>>> sumprimes([3,3,1,13])
19
>>> sumprimes([2,4,6,9,11])
13
>>> sumprimes([-3,1,6])
0
```

Private Test cases used for evaluation	**Input**	**Expected Output**	**Actual Output**	**Status**
Test Case 1	`intreverse(31511)`	`11513\n`	`11513\n`	Passe
Test Case 2	`intreverse(4)`	`4\n`	`4\n`	Passe
Test Case 3	`intreverse(15135324234235)`	`53243242353151\n`	`53243242353151\n`	Passe
Test Case 4	`matched("a3qw3;4w3(aasdgsd) ((agadsgdsgag)agaga)")`	`True\n`	`True\n`	Passe
Test Case 5	`matched(" (ag(Gaga(agag)Gaga)GG)a)33)cc(")`	`False\n`	`False\n`	Passe
Test Case 6	`matched("(adsgdsg(agaga)a")`	`False\n`	`False\n`	Passe
Test Case 7	`matched("15ababa.agaga[][[[")`	`True\n`	`True\n`	Passe
Test Case 8	`sumprimes([101,93,97,44])`	`198\n`	`198\n`	Passe
Test Case 9	`sumprimes([1001,393,743,59])`	`802\n`	`802\n`	Passe
Test Case 10	`sumprimes([11,11,11,13,11,-11])`	`57\n`	`57\n`	sse

Text Transcripts ()

Books ()

Download Videos ()

Problem Solving Session ()

The due date for submitting this assignment has passed.

10 out of 10 tests passed.

You scored 100.0/100.

Assignment submitted on 2022-08-11, 20:00 IST

Your last recorded submission was :

```
# intraverse function for integers
def intreverse(n):
  reversed_n = 0
  while n > 0:
    (d, n) = (n%10, n//10)
    reversed_n = 10*reversed_n + d
  return(reversed_n)

## matching string function
def matched(s):
  nested = 0
  for i in range(0, len(s)):
    if s[i] == "(":
      nested = nested + 1
    elif s[i] == ")":
      nested = nested - 1
      if nested < 0:
        return(False)
  return(nested == 0)


## factoring integer

def factor(n):
  factorlist = []
  for i in range(1,n+1):
    if n%i == 0:
      factorlist.append(i)
  return(factorlist)

## to check prime number

def isprime(n):
  return(factor(n) == [1, n])

# sum of primes

def sumprimes(l):
  sum = 0
  for i in range(0,len(l)):
    if isprime(l[i]):
      sum = sum + int(l[i])
  return(sum)
import ast

def tolist(inp):
  inp = "["+inp+"]"
  inp = ast.literal_eval(inp)
  return (inp[0],inp[1])

def parse(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 == "intreverse":
   arg = parse(farg)
   print(intreverse(arg))
elif fname == "matched":
   arg = parse(farg)
   print(matched(arg))
elif fname == "sumprimes":
   arg = parse(farg)
   print(sumprimes(arg))
else:
   print("Function", fname, "unknown")


```

Sample solutions (Provided by instructor)

```python
def intreverse(n):
  ans = 0
  while n > 0:
    (d,n) = (n%10,n//10)
    ans = 10*ans + d
  return(ans)

def matched(s):
  nested = 0
  for i in range(0,len(s)):
    if s[i] == "(":
       nested = nested+1
    elif s[i] == ")":
       nested = nested-1
       if nested < 0:
          return(False)
  return(nested == 0)

def factors(n):
  factorlist = []
  for i in range(1,n+1):
    if n%i == 0:
      factorlist = factorlist + [i]
  return(factorlist)

def isprime(n):
  return(factors(n) == [1,n])


def sumprimes(l):
  sum = 0
  for i in range(0,len(l)):
    if isprime(l[i]):
      sum = sum+l[i]
  return(sum)


import ast

def tolist(inp):
  inp = "["+inp+"]"
  inp = ast.literal_eval(inp)
  return (inp[0],inp[1])

def parse(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 == "intreverse":
   arg = parse(farg)
   print(intreverse(arg))
elif fname == "matched":
   arg = parse(farg)
   print(matched(arg))
elif fname == "sumprimes":
   arg = parse(farg)
   print(sumprimes(arg))
else:
   print("Function", fname, "unknown")


```