

X



(<https://swayam.gov.in>)



([https://swayam.gov.in/nc\\_details/NPTEL](https://swayam.gov.in/nc_details/NPTEL))

[gauravsharma727545@gmail.com](mailto: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](https://swayam.gov.in/nd1_noc19_cs40/preview))** [Ask a Question \(forum\)](#)

[Progress \(student/home\)](#) [Mentor \(student/mentor\)](#)

# Week 2 Programming Assignment

**Due on 2019-08-22, 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.

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 2  
Programming  
Assignment**  
(/noc19\_cs40/progassignment?  
name=90)

**Week 3: Lists, inductive function 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**

```
>>> 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 matched brackets and `False` if it does not.

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 returns 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	Pas sed
Test Case 2	intreverse(4)	4\n	4\n	Pas sed

**Week 7 Quiz****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**

Test Case 3

`intreverse(15135324  
234235)``53243  
242353  
151\n``53243  
24235  
3151  
\n`Pas  
sed

Test Case 4

`matched("a3qw3;4w3  
(aasdgsd)((agadsgds  
gag)agaga")``True  
\n``True  
\n`Pas  
sed

Test Case 5

`matched("(ag(Gaga(a  
gag)Gaga)GG)a)33)cc  
(")``False  
\n``False  
\n`Pas  
sed

Test Case 6

`matched("(adsgdsg(a  
gaga)a")``False  
\n``False  
\n`Pas  
sed

Test Case 7

`matched("15ababa.ag  
aga[[[")``True  
\n``True  
\n`Pas  
sed

Test Case 8

`sumprimes([101,93,9  
7,44])``198\n``198\n`Pas  
sed

Test Case 9

`sumprimes([1001,39  
3,743,59])``802\n``802\n`Pas  
sed

Test Case 10

`sumprimes([11,11,1  
1,13,11,-11])``57\n``57\n`Pas  
sed

Due Date Exceeded.

10 out of 10 tests passed.

You scored 100.0/100.

Your last recorded submission was :

```
1 #program for reversing a number
2 def intreverse(n):
3     num = str(n)
4     num = num[::-1]
5     return num
6
7
8 #program for matching brackets in given string
9 def matched(s):
10     counter = 0
11     for bracket in s:
12         if bracket == ')':
13             counter -= 1
14             if counter < 0:
15                 return False
16
17         elif bracket == '(':
18             counter += 1
19     return counter == 0
20
21
22 #program for sum of prime in a given list of numbers
```

```

23 def sumprimes(l):
24
25     def factor(n):
26         factors = []
27         for i in range(1,n+1):
28             if n%i == 0:
29                 factors.append(i)
30         return factors
31
32     def isprime(n):
33         f = factor(n)
34         if f == [1,n]:
35             return True
36
37     sum = 0
38     for num in l:
39         if isprime(num):
40             sum += num
41     return sum
42 import ast
43
44 def tolist(inp):
45     inp = "["+inp+"]"
46     inp = ast.literal_eval(inp)
47     return (inp[0],inp[1])
48
49 def parse(inp):
50     inp = ast.literal_eval(inp)
51     return (inp)
52
53 fncall = input()
54 lparen = fncall.find("(")
55 rparen = fncall.rfind(")")
56 fname = fncall[lparen]
57 farg = fncall[lparen+1:rparen]
58
59 if fname == "intreverse":
60     arg = parse(farg)
61     print(intreverse(arg))
62 elif fname == "matched":
63     arg = parse(farg)
64     print(matched(arg))
65 elif fname == "sumprimes":
66     arg = parse(farg)
67     print(sumprimes(arg))
68 else:
69     print("Function", fname, "unknown")
70
71

```

Sample solutions (Provided by instructor)

Select the Language . Python3 ▼

```

1 def intreverse(n):
2     ans = 0
3     while n > 0:
4         (d,n) = (n%10,n//10)
5         ans = 10*ans + d
6     return(ans)
7
8 def matched(s):
9     nested = 0
10    for i in range(0,len(s)):
11        if s[i] == "(":
12            nested = nested+1
13        elif s[i] == ")":
14            nested = nested-1
15            if nested < 0:
16                return(False)
17    return(nested == 0)
18
19 def factors(n):
20     factorlist = []
21     for i in range(1,n+1):
22         if n%i == 0:

```

```

23     factorlist = factorlist + [i]
24     return(factorlist)
25
26 def isprime(n):
27     return(factors(n) == [1,n])
28
29
30 def sumprimes(l):
31     sum = 0
32     for i in range(0,len(l)):
33         if isprime(l[i]):
34             sum = sum+l[i]
35     return(sum)
36
37
38 import ast
39
40 def tolist(inp):
41     inp = "["+inp+"]"
42     inp = ast.literal_eval(inp)
43     return (inp[0],inp[1])
44
45 def parse(inp):
46     inp = ast.literal_eval(inp)
47     return (inp)
48
49 fncall = input()
50 lparen = fncall.find("(")
51 rparen = fncall.rfind(")")
52 fname = fncall[:lparen]
53 farg = fncall[lparen+1:rparen]
54
55 if fname == "intreverse":
56     arg = parse(farg)
57     print(intreverse(arg))
58 elif fname == "matched":
59     arg = parse(farg)
60     print(matched(arg))
61 elif fname == "sumprimes":
62     arg = parse(farg)
63     print(sumprimes(arg))
64 else:
65     print("Function", fname, "unknown")
66
67

```