simpson_1_3_rule.py

```
#-------------------------------------------------------------------------------
# Name: simpson_1_3_rule
# Purpose: compute definite integral
#
# Author: Abdul Qadir Lonara
#
# Created:
# Copyright: (c) Abdul Qadir Lonara 2023
# Licence: <your licence>
#-------------------------------------------------------------------------------
def simpson_1_3_rule(f, a, b, n):
"""
Calculates the definite integral of a function f using Simpson's 1/3rd rule.
Arguments:
f -- the function to integrate
a -- the lower limit of integration
b -- the upper limit of integration
n -- the number of intervals to use
Returns:
The value of the definite integral of f from a to b using Simpson's 1/3rd rule.
"""
if n % 2 != 0:
raise ValueError("n must be an even integer")
h = (b - a) / n
x = [a + i * h for i in range(n+1)]
integral = f(x[0])
for i in range(1, n):
if i % 2 == 0:
integral += 2 * f(x[i])
else:
integral += 4 * f(x[i])
integral += f(x[-1])
integral *= h / 3
return integral
def f(x):
return eval(fun(x))
y = input("Enter function : ")
```

```
fun = lambda x : eval(y)

a = float(input("Enter the lower limite a : "))

b = float(input("Enter the upper limite b : "))

n = int(input("Enter the number of intervals to use :"))

integral = simpson_1_3_rule(fun, a, b, n)

print("The value of the definite integral of f from a to b using Simpson's 1/3rd rule = "+ str(integral))
```

Output :-

```
Enter function : 1/(x**2+6*x+10)

Enter the lower limite a : 0

Enter the upper limite b : 1

Enter the number of intervals to use :4

The value of the definite integral of f from a to b using Simpson's 1/3rd rule = 0.07677275312993578
```