Отчет по домашнему заданию на 26.01.18

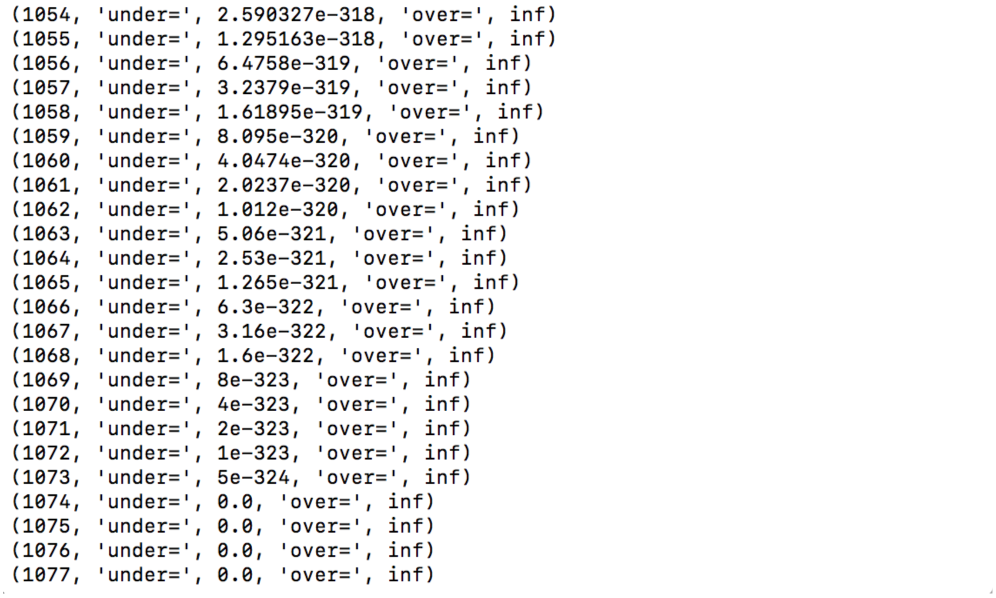
Выполнил: Ширшов Александр Юрьевич

Если учесть, что за каждое задание 1 балл, то желаемый балл 11.

Напишите программу, которая определяет пределы нижнего и верхнего переполнения (в степенях 2) дляPython на вашем компьютере.

```
N=1100

under=1.

over=1.

for i in range (N):

        under=under / 2.

        over=over*2.

        print(i,'under=' , under, 'over=' , over)
```

Получаем результат



```
(1054, 'under=', 2.590327e-318, 'over=', inf)
(1055, 'under=', 1.295163e-318, 'over=', inf)
(1056, 'under=', 6.4758e-319, 'over=', inf)
(1057, 'under=', 3.2379e-319, 'over=', inf)
(1058, 'under=', 1.61895e-319, 'over=', inf)
(1059, 'under=', 8.095e-320, 'over=', inf)
(1060, 'under=', 4.0474e-320, 'over=', inf)
(1061, 'under=', 2.0237e-320, 'over=', inf)
(1062, 'under=', 1.012e-320, 'over=', inf)
(1063, 'under=', 5.06e-321, 'over=', inf)
(1064, 'under=', 2.53e-321, 'over=', inf)
(1065, 'under=', 1.265e-321, 'over=', inf)
(1066, 'under=', 6.3e-322, 'over=', inf)
(1067, 'under=', 3.16e-322, 'over=', inf)
(1068, 'under=', 1.6e-322, 'over=', inf)
(1069, 'under=', 8e-323, 'over=', inf)
(1070, 'under=', 4e-323, 'over=', inf)
(1071, 'under=', 2e-323, 'over=', inf)
(1072, 'under=', 1e-323, 'over=', inf)
(1073, 'under=', 5e-324, 'over=', inf)
(1074, 'under=', 0.0, 'over=', inf)
(1075, 'under=', 0.0, 'over=', inf)
(1076, 'under=', 0.0, 'over=', inf)
(1077, 'under=', 0.0, 'over=', inf)
```

Проверьте, где происходит under- и overflow для чисел с плавающей запятой(floats).

```
N=1100
under=float(1.)
over=float(1.)
for i in range (N):
        over=over * 2
        print(i,'under=' , under, 'over=' , over)
```

Ответ:

```
(1022, 'under=', 1.0, 'over=', 8.98846567431158e+307)
```

Проверьте, где происходит under- и overflow для чисел с плавающей запятойс двойной точностью.

```
N=20
over=int(1)
for i in range (N):
        over=over * 2
        print(i-1, 'over=' , over)
```

Ответ: (18, 'over=', 1048576)

Задача 3

а)

```
eps = 1.0
one_Plus_eps = 2.0
while (one_Plus_eps != 1):
        eps = eps /2
        one_Plus_eps = 1.0 + eps
        print ( " eps = " , eps , " , one + eps = " , one_Plus_eps )
```

Для типа float ε ≈ 1 · 10−16

б)

```
eps = complex(1.0, 0.0)
one_Plus_eps = (2.0, 0.0)
while (one_Plus_eps != complex(1.0, 0.0)):
        eps = eps /2
        one_Plus_eps = 1.0 + eps
        print ( " eps = " , eps , " , one + eps = " , one_Plus_eps )
```

Для типа complexε ≈ 1 · 10−16

Задача 4

1)

```
 import math
def my_sin(x):
term = x
sum = x
eps = 10**(-8)
n = 1
while (abs(term) > abs(sum * eps)):
n += 1
term = -term * x**2 /( (2*n-1) * (2*n-2) )
sum = sum + term
return sum
while (True):
try:
x = float(input())
except:
```

```
print ("Exit")
break
if (math.sin(x) != 0):
print (abs ((my_sin(x) - math.sin(x))/(math.sin(x))))
else:
print ("sin(x) = 0, my_sin(x) = ", my_sin(x))
```

2)

x	εsin(x)
0 (0)	0.0/0.0
3.14159 (π)	4 · 10−11
3.141592654 (π)	8 · 10−7
6.28318 (2π)	2 · 10−10
6.283185307 (2π)	6 · 10−6
1.57 (π/2)	6 · 10−12
4.71 (3π/2)	6 · 10−12
31.4159 (10π)	7 · 100
31.415926 (10π)	1 · 103
314.159 (100π)	6 · 10121
314.1592654 (3π/2)	2 · 10126
32.99 (10π + π/2)	4 · 10−4
32.98672286 (10π + π/2)	8 · 10−4
315.73 (100π + π/2)	5 · 10118
315.7300617 (100π + π/2)	2 · 10119

13)

```
import math
def my_sin(x):
while (abs(x) > math.pi):
x -= 2*math.pi
term = x
sum = x
```

```
eps = 10**(-8)
n = 1
while (abs(term) > abs(sum * eps)):
n += 1
term = -term * x**2 /( (2*n-1) * (2*n-2) )
sum = sum + term
return sum
while (True):
try:
x = float(input())
except:
print ("Exit")
break
if (math.sin(x) != 0):
print (abs ((my_sin(x) - math.sin(x))/(math.sin(x))), my_sin(x +
2*math.pi) - my_sin(x))
else:
print ("sin(x) = 0, my_sin(x) = ", my_sin(x))
```

Задача 5

```
from math import *
def solve(a, b, c):
if (b**2 >= 4*a*c) and (a != 0) and (c != 0):
print ((-b + sqrt(b**2-4*a*c))/(2*a), (-b - sqrt(b**2-4*a*c))/(2*
a))
print ((2*c)/(-b - sqrt(b**2-4*a*c)), (2*c)/((-b + sqrt(b**2-4*a*
c))))
sum = 0
x = -4*a*c/b**2
f = 1
term = 1
```

```
sum = 1
eps = 10 ** (-40)
n = 0
while (abs(term) > abs(sum * eps)):
term = term * x * (1-2*n)/(2*(n+1))
sum = sum + term
n += 1
print (b*(-1+b/abs(b)*sum)/(2*a), b*(-1-b/abs(b)*sum)/(2*a))
print ((2*c)/(b*(-1-b/abs(b)*sum)), (2*c)/(b*(-1+b/abs(b)*sum)))
else:
if (a == 0) and ( b != 0):
print ((-c/b), (-c/b))
else:
if (a == 0) and ( b == 0):
if (c == 0):
print ("x in (-inf, +inf)")
else:
if ( c != 0):
print("NaN, NaN")
else:
```

```
print("Error: D < 0")
while (True):
try:
a = float(input("a = "))
b = float(input("b = "))
c = float(input("c = "))
except:
print ("Exit")
break
solve(a, b, c)
```

Задача 6

а)

```
from math import *
def s_1(N):
n = 1
sum = -0.5
term = -0.5
while (n < 2*N):
term = term * (-1)*((n+1)**2)/(n*(n+2))
sum += term
n += 1
return sum
def s_2(N):
n = 1
sum = 0
while (n < N):
sum += -(2*n-1)/(2*n) + (2*n)/(2*n+1)
n += 1
return sum
def s_3(N):
n = 1
sum = 1/6
term = 1/6
while (n < N):
term = term * (2*n*(2*n+1))/((2*n+2)*(2*n+3))
sum += term
n += 1
return sum
```

б)

```
from math import *

from tkinter import *

def s_1(N):

n = 1

sum = -0.5

term = -0.5

while (n < 2*N):

term = term * (-1)*((n+1)**2)/(n*(n+2))

sum += term

n += 1

return sum

def s_2(N):

n = 1

sum = 0

while (n < N):

sum += -(2*n-1)/(2*n) + (2*n)/(2*n+1)

n += 1

return sum

def s_3(N):

n = 1

sum = 1/6

term = 1/6

while (n < N):

term = term * (2*n*(2*n+1))/((2*n+2)*(2*n+3))

sum += term

n += 1

return sum

def err(x):

try:

return abs((s_1(x)-s_3(x))/s_3(x))
```

```
except:
return 1
root = Tk()
x_0 = 10**(0)
y_0 = 10**(2)
x_sc = 10**(0)
canv = Canvas(root, width=1000, height=1000, bg="white")
canv.create_line(500, 1000, 500, 0, width=2, arrow=LAST)
canv.create_line(0, 500, 1000, 500, width=2, arrow=LAST)
canv.create_text(980, -20 + 500, font=("Purisa", 18), text="x", fill="
purple")
canv.create_text(-57 + 500, 25, font = ("Purisa", 15), text="err*"+ str(
x_sc), fill="purple")
First_x = -500;
my_file = open("lg.dat", "w")
for i in range(18000):
if (i % 1800 == 0):
k = First_x + (1 / 18) * i
canv.create_line(k + 500, -3 + 500, k + 500, 3 + 500, width=0.5,
fill='black')
canv.create_line(k + 500, 0, k + 500, 1000, width=0.1, fill='grey
', dash=(1, 1))
canv.create_text(k + 515, 10 + 500, font = ("Purisa", 10), text=
str(k/x_0), fill="purple")
if (k != 0):
canv.create_line(-3 + 500, k + 500, 3 + 500, k + 500, width
=0.5, fill='black')
canv.create_line(0, k + 500, 1000, k + 500, width=0.1, fill='
grey', dash=(1, 1))
canv.create_text(25 + 500, k + 500 + 20, font = ("Purisa",
10), text=str(-k/y_0*x_sc), fill="purple")
```

```
try:
x = First_x + (1 / 18) * i
y = -err(x/x_0)*y_0 + 499
x += 499
canv.create_oval(x, y, x + 1, y + 1, fill='black')
my_file.write(str(x - 499) + " " + str(499-y) + "\n")
except:
First_x = -500
my_file.close()
canv.pack()
root.mainloop()
```