**Исламов Радмир Лабораторная 1**

**Задание 1**

Подставим значение p=poly(1:20) в функцию roots(p) для нахождения корней полинома:  
>> p=poly(1:20)

roots(p)

ans =

20.0003

18.9972

18.0112

16.9711

16.0483

14.9354

14.0653

12.9491

12.0334

10.9840

10.0061

8.9984

8.0003

7.0000

6.0000

5.0000

4.0000

3.0000

2.0000

1.0000

Корни полинома найдены верно.

Теперь изменим входное значение на малую величину:

>> p(2)=p(2) + 10^-7

roots(p)

ans =

-0.3064 + 1.4438i

-0.3064 - 1.4438i

-0.0036 + 0.2407i

-0.0036 - 0.2407i

0.0358 + 0.1261i

0.0358 - 0.1261i

0.0453 + 0.0779i

0.0453 - 0.0779i

0.0474 + 0.0501i

0.0474 - 0.0501i

0.0466 + 0.0314i

0.0466 - 0.0314i

0.0444 + 0.0177i

0.0444 - 0.0177i

0.0413 + 0.0071i

0.0413 - 0.0071i

0.0383

0.0300

0.0200

0.0100

Половина значений оказались комплексными. Исходная задача оказалась неустойчива к входным данным, результат абсурдный.

Это ошибка входных данных.

**Задание 2**

>> 2^1023

2^2024

realmax

realmin

ans =

8.9885e+307

ans =

Inf

ans =

1.7977e+308

ans =

2.2251e-308

Мы получаем Inf так как выходим за пределы представления числа

**Задание 3**

>> format long e

sqrt(2)

ans =

1.414213562373095e+000

Получили всего 16 цифр и 15 в мантисе. Всё что выходит за предел в 15-16 цифр для long e, отбрасывается.

**Задание 4**

Казалось бы подобной точности более чем достаточно, но порой даже такая точность создает ошибки в вычислениях.

>> format long e

10^8+10^-7

10^8+10^-8

ans =

1.000000000000001e+008

ans =

1.000000000000000e+008

Как мы видим ответ отличается, хотя суммируемые числа одинаковы.

**Задание 5**

>> format long e

x=1;

for c=1:10^6

x = x+10^-16;

end

disp(x)

x=1+(10^-16\*10^17);

disp(x)

>> Ex5

1

11

Как видим ошибка 1000%. Желательно сперва считать малые числа, а потом большие.

Задание 6

>> n=1;

e=@(n)2^-n;

>> while((1+e(n)-1)/(e(n)==1))

n=n+1;

end

>> n

n =

53

Верно на 53 шаге отличается от еденицы

**Задание 7**

>> n=1;

f=@(x,n)x^n\*exp(x-1);

syms x

for n=1:30

n=n+1; int(f(x,n),x,0,1)

end

format long e

n=1;

f=@(x,n)x^n\*exp(x-1);

syms x

for n=1:30

n=n+1;

subs(int(f(x,n),x,0,1))

end

ans =

2.642411176571153e-001

ans =

2.072766470286539e-001

ans =

1.708934118853843e-001

ans =

1.455329405730786e-001

ans =

1.268023565615284e-001

ans =

1.123835040693008e-001

ans =

1.009319674455933e-001

ans =

9.161229298966059e-002

ans =

8.387707010339417e-002

ans =

7.735222886266420e-002

ans =

7.177325364802956e-002

ans =

6.694770257561571e-002

ans =

6.273216394138015e-002

ans =

5.901754087929777e-002

ans =

-5.764607523034235e+017

ВЫШЛО ОТРИЦАЬЕЛЬНОЕ ЗНАЧЕНИЕ

ans =

5.277111916899476e-002

ans =

5.011985495809426e-002

ans =

4.772275579620910e-002

ans =

4.554488407581805e-002

ans =

4.355743440782089e-002

ans =

4.173644302794031e-002

ans =

4.006181035737290e-002

ans =

3.851655142305043e-002

ans =

3.708621442373924e-002

ans =

3.575842498277981e-002

ans =

3.452252546494532e-002

ans =

3.336928698153123e-002

ans =

3.229067753559440e-002

ans =

3.127967393216808e-002

ans =

3.033010810278951e-002

У нас накопилась ошибка выдающая отрицательное значение подынтегрального выражения

**Задание 8**

function [res]=sinRyad(x)

res = 0; i = 0;

while (x^(2\*i + 1)/factorial(2\*i + 1) > 10^-17)

res = res + (-1)^i\*x^(2\*i+1)/factorial(2\*i + 1);

i = i + 1;

end

end

>> Ex7(0), Ex7(pi/3), Ex7(pi/2), Ex7(pi), Ex7(2\*pi)

ans =

0

ans =

0.8660

ans =

1.0000

ans =

3.3312e-016

ans =

3.3012e-015

Значениея достаточно близко равны к полагаемым

>> Ex7(12\*pi), Ex7(13\*pi), Ex7(14\*pi)

ans =

0.1140

ans =

13.1806

ans =

445.1056

Данные значения объясняются погрешностью округления. Лучше приводить значения к отрезку [0; pi/2]