

Задание 3

0. Задание (сделайте себе шпаргалку перед глазами, если не помните) - не присылать

Чему равны синус, косинус, тангенс перечисленных углов?

Запишите значения в таблицу:

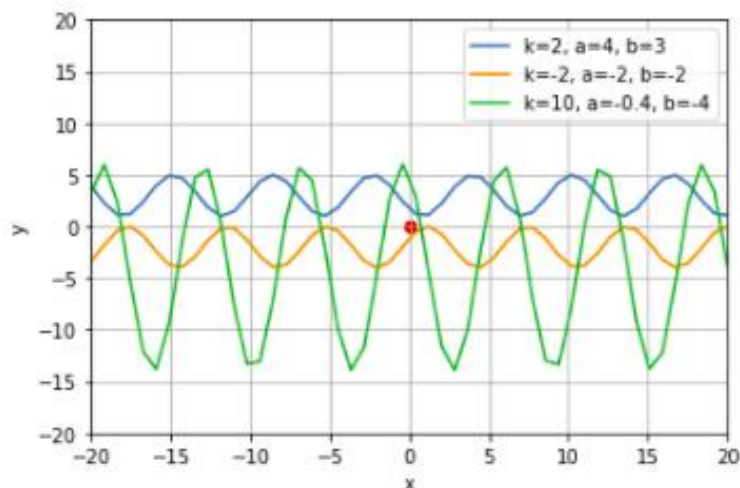
угол	sin	cos	tg
30°	1/2	$\sqrt{3}/2$	$\sqrt{3}/3$
45°	$\sqrt{2}/2$	$\sqrt{2}/2$	1
60°	$\sqrt{3}/2$	1/2	$\sqrt{3}$
90°	1	0	-
180°	0	-1	0

1. Задание (в программе)

Нарисуйте график функции: $y(x) = k \cdot \cos(x - a) + b$ для некоторых (2-3 различных) значений параметров k , a , b

```
x = np.linspace(-20, 20)

fig, ax = plt.subplots()
plt.ylim(-20, 20)
plt.xlim(-20, 20)
ax.plot(x, 2*cos(x - 4) + 3, label = 'k=2, a=4, b=3')
plt.plot(x, -2*cos(x + 2) - 2, label = 'k=-2, a=-2, b=-2')
plt.plot(x, 10*cos(x + 0.4) - 4, label = 'k=10, a=-0.4, b=-4')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.scatter(0, 0, color='r')
ax.legend()
```



2. Задание

Докажите, что при ортогональном преобразовании сохраняется расстояние между точками.

При ортогональном преобразовании плоскости скалярное преобразование сохраняется в любом ортонормированном базисе. Из сохранения при ортогональном преобразовании скалярного произведения для любой пары векторов следует сохранение длин векторов

3. Задание (в программе)

1. Напишите код, который будет переводить полярные координаты в декартовы.

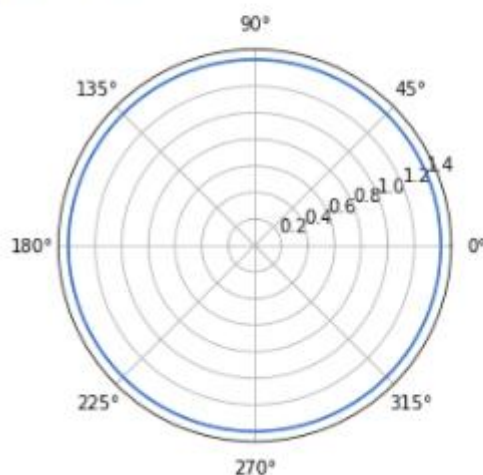
```
#1.*Напишите код, который будет переводить полярные координаты в декартовы.  
  
def polar_to_dec(x,y):  
    R = sqrt(x**2 + y**2)  
    A = np.arctan2(y,x)  
    X = R*np.cos(A)  
    Y = R*np.sin(A)  
  
    print(f'R = ', round(R, 2))  
    print(f'A = ', round(A, 2), 'рад')  
    print(f'Полярные координаты: (', round(A*180/np.pi, 2),',',round(R, 2),')')  
    print(f'Проверка! Изначальные координаты точки:', '(',int(X),',',int(Y),')')
```

```
polar_to_dec(5,6)
```

```
R = 7.81  
A = 0.88 рад  
Полярные координаты: ( 50.19 , 7.81 )  
Проверка! Изначальные координаты точки: ( 5 , 6 )
```

2. Напишите код, который будет рисовать график окружности в полярных координатах.

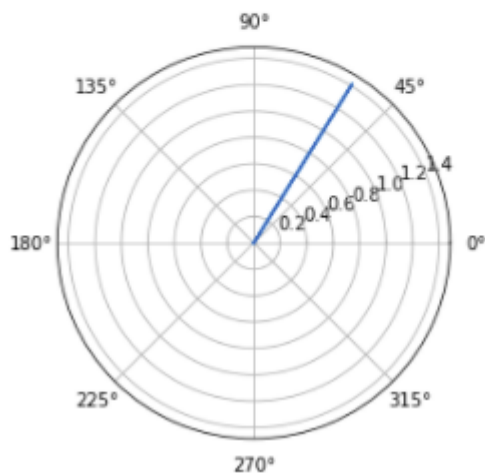
```
theta = np.arange(0., 2., 1./180.)*np.pi # 360 точек  
plt.polar(theta, [1.4]*len(theta)) # окружность  
plt.show()
```



Напишите код, который будет рисовать график отрезка прямой линии в полярных координатах.

```
t=np.linspace(0,np.pi,1000)
x=np.sin(t)
y=np.sin(t)
R=np.sqrt(x**2+y**2)
phi=np.arctan2(y,x)*180/np.pi
plt.polar(phi,R)
```

[<matplotlib.lines.Line2D at 0x20a4ab14280>]



4. Задание (в программе)

$\exp(x) + x \cdot (1 - y) = 1$ Решите систему уравнений:

$$y = x^2 - 1$$

$$\exp(x) + x - x \cdot y - 1 = 0$$

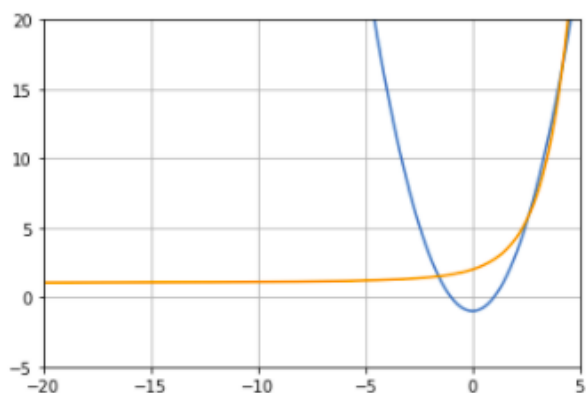
$$y - x^2 + 1 = 0$$

```
from scipy.optimize import fsolve
```

```
#exp(x) + x*(1 - y) = 1
#y = x^2 - 1

x = np.linspace(-20, 20, 1000)
plt.grid(True)
plt.ylim(-5, 20)
plt.xlim(-20, 5)
plt.plot(x, x**2-1)
plt.plot(x, (np.exp(x)+x-1)/x)
```

[<matplotlib.lines.Line2D at 0x231e8dee460>]



```
def slu_solve(p):  
    x,y = p  
    return(np.exp(x)+x-x*y-1, y-x**2+1)  
x1,y1 = fsolve(slu_solve,(-5,0))  
print(x1,y1)
```

-1.581835352895898 1.5022030836712905

1)

2) Решите систему уравнений и неравенств:

$$y = x^2 - 1$$

$$\exp(x) + x \cdot (1 - y) - 1 > 0$$

я не поняла, как решить систему с неравенством(((