

```

from typing import get_origin
import matplotlib.pyplot as plt
from operator import truediv
import numpy as np
from numpy.core.einsumfunc import einsum
from numpy.core.fromnumeric import product
from numpy.lib.function_base import diff

def menu():
    print('choose your operation!')
    print('1.Enter vector a')
    print('2.Enter vector b')
    print('3.Calculate a + b')
    print('4.Calculate a - b')
    print('5.Multiply a by scalar (constant)')
    print('6.Calculate scalar product between a and b')
    print('7.Calculate vector product between a and b')
    print('8.Enter plan p')
    print('9.Show plan p')
    print('10.Project the vector a on the plane p')
    print('0.Exit the program')

def vectors():
    stop=False
    while not stop:
        try:
            vector=list(map(float,input('Enter vector value (x,y,z) on the
mold x y z: ').split()))
            if len(vector)<3:
                raise ValueError
        except ValueError as error:
            print('invalid entry!,try again!')
        else:
            stop=True
            return vector

def the_plan():
    stop=False
    while not stop:
        try:
            plan=list(map(float,input('Enter the desired parameters on the
form a b c d: ').split()))
            if len(plan)<4:
                raise ValueError
        except ValueError as error:
            print('invalid entry!,try again!')
        else:
            stop=True
            return plan

def Project_vector1_on_plane_p(vector_a,n_vector):
    scalar=np.dot(vector_a,n_vector)
    length=np.array(list(map(lambda x:x**2,n_vector)))
    sum_length=np.sum(length)

```

```

product=np.array(list(map(lambda x:x*(scalar/sum_length),n_vector)))
project= vector_a-product

return project

def
graph_vector(END_POINT,color1,END_POINT1,color2,END_POINT2,color3,filename
,x):
    ORIGIN=[0,0,0]

    myfig = plt.figure()
    axes = myfig.gca(projection='3d')
    axes.set_title("Exempel på vector")
    axes.set_xlabel('X')
    axes.set_ylabel('Y')
    axes.set_zlabel('Z')
    axes.set_xlim(-x, x)
    axes.set_ylim(-x, x)
    axes.set_zlim(-x, x)
    plt.quiver(*ORIGIN, *END_POINT, color=color1)
    plt.quiver(*ORIGIN, *END_POINT1,color=color2)
    plt.quiver(*ORIGIN,*END_POINT2,color=color3)
    plt.show()
    myfig.savefig(filename)

    plt.close(myfig)
    return myfig

def graph_vector_2D(END_POINT,color1,filename,x,END_POINT1,color3):
    ORIGIN=[0,0,0]
    myfig = plt.figure()
    axes = myfig.gca(projection='3d')
    axes.set_title("Exempel på vector")
    axes.set_xlabel('X')
    axes.set_ylabel('Y')
    axes.set_zlabel('Z')
    axes.set_xlim(-x, x)
    axes.set_ylim(-x, x)
    axes.set_zlim(-x, x)
    plt.quiver(*ORIGIN, *END_POINT, color=color1)
    plt.quiver(*ORIGIN, *END_POINT1, color=color3)
    plt.show()
    myfig.savefig(filename)

    plt.close(myfig)
    return myfig

def graph_plan(lista):
    a,b,c,d=lista
    filename = "plane.png"
    if a==0 and c==0:
        x,z=np.meshgrid(range(11),range(11))
        y=-(a*x + c*z + d) / b
    elif b==0 and c==0:

```

```

        y,z=np.meshgrid(range(11),range(11))
        x=-(b*y+c*z + d) / a
    elif c==0 :
        x,z=np.meshgrid(range(11),range(11))
        y=-(a*x+c*z+d)/b
    else:
        x,y=np.meshgrid(range(11),range(11))
        z=-(a*x + b*y + d) / c
    myfig=plt.figure()
    axes=myfig.gca(projection='3d')
    axes.plot_surface(x, y, z)
    axes.set_title("Exempel på plane")
    axes.set_xlabel('X')
    axes.set_ylabel('Y')
    axes.set_zlabel('Z')
    plt.show()
    myfig.savefig(filename)
    plt.close(myfig)
    return myfig

def graph_project(lista,vector_a,n_vector,project):
    ORIGIN=[0,0,0]
    a,b,c,d=lista
    filename = "plane.png"
    x,y=np.meshgrid(range(11),range(11))
    z=-(a*x + b*y + d) / c
    myfig=plt.figure()
    axes=myfig.gca(projection='3d')
    axes.plot_surface(x, y, z)
    axes.set_title("Exempel på plane")
    axes.set_xlabel('X')
    axes.set_ylabel('Y')
    axes.set_zlabel('Z')
    plt.quiver(*ORIGIN, *vector_a, color='green')
    plt.quiver(*ORIGIN, *n_vector,color='red')
    plt.quiver(*ORIGIN,*project,color='black')
    plt.show()
    myfig.savefig(filename)
    plt.close(myfig)
    return myfig

def main():
    print('Welcome to the graphical algebra calculator')
    print('_____')
    print('_____')
    vector_a=[]
    vector_b=[]
    plan=[]
    n_vector=[]
    stop=True
    while stop==True:
        print('_____')
        print('_____')
        choice=menu()

```

```

try:
    choice=int(input('Your choice: '))
    if (choice<0) or (choice>10):
        raise ValueError
except ValueError as error:
    print('invalid entry!,try again!')

if choice==0:
    stop=False
    print('program exits ...')

elif choice==1:
    print('Takes values  $\hat{a}$  for vector a!')
    vector_a=vectors()

elif choice==2:
    print('Takes values  $\hat{b}$  for vector b!')
    vector_b=vectors()

elif choice==3:
    vec1=np.array(vector_a)
    vec2=np.array(vector_b)
    try:
        sum=vec1+vec2
        if not vector_a and not vector_b:
            raise ValueError
    except ValueError :
        print('OBS!,You miss one of vectors,Try again')
    else:
        max_value=vector_a+vector_b+list(sum)
        list_abs=list(map(lambda x:abs(x),max_value))
        print(vector_a,'+',vector_b,'=',sum)
        print('_____')
        print('Do you want to save the image to file?')
        print('(Answer Y to save, other choices mean no.)')
        choice_image=input('Your choies: ')
        if choice_image=='Y':
            filnam=input('filename: ')
            format='.png'
            filename=filnam + format
            graph_vector(sum,'green',vector_a,'red',vector_b,'blue
',filename,max(list_abs))
            print('Do you want to save the result in a or b?')
            print('(Answer a or b to save, other choices mean no.)')
            choies=input('Your choies: ')
            if choies == 'a':
                vector_a=list(sum)
            elif choies == 'b':
                vector_b=list(sum)

elif choice==4:
    vec1=np.array(vector_a)
    vec2=np.array(vector_b)
    try:

```

```

        differins=vec1-vec2
        if not vector_a and not vector_b:
            raise ValueError
except ValueError:
    print('OBS!,You miss one of vectors,Try again')
else:
    max_value=vector_a+vector_b+list(differins)
    list_abs=list(map(lambda x:abs(x),max_value))
    print(vector_a,'-',vector_b,'=',differins)
    print('_____')
    print('Do you want to save the image to file?')
    print('(Answer Y to save, other choices mean no.)')
    choice_image=input('Your choies: ')
    if choice_image=='Y':
        filnam=input('filename: ')
        format='.png'
        filename=filnam + format
        graph_vector(differins,'green',vector_a,'red',vector_b
, 'blue',filename,max(list_abs))
        print('Do you want to save the result in a or b?')
        print('(Answer a or b to save, other choices mean no.)')
        choies=input('Your choies: ')
        if choies == 'a':
            vector_a=list(differins)
        elif choies == 'b':
            vector_b=list(differins)

elif choice==5:
    stop1=True
    while stop1==True:
        if vector_a==[]:
            print('OBS!,You miss one of vectors,Try again')
            stop1=False
        else:
            try:
                scalar=float(input('Enter constant: '))
                calculate=list(map(lambda x:x*scalar,vector_a))
            except ValueError as error:
                print('invalid entry!,try again!')
            else:
                stop1=False
                print(vector_a,'*',scalar,'=',calculate)

                print('_____')

                print('Do you want to save the image to file?')
                print('(Answer Y to save, other choices mean
no.)')

                choice_image=input('Your choies: ')
                if choice_image=='Y':
                    max_value=vector_a+list(calculate)
                    list_abs=list(map(lambda x:abs(x),max_value))
                    filnam=input('filename: ')
                    format='.png'

```

```

        filename=filnam + format
        graph_vector_2D(calculate,'green',filename,max(
list_abs),vector_a,'red')
        print('Do you want to save the result in a or b?')
        print('(Answer a or b to save, other choices mean
no.)')

        choies=input('Your choies: ')
        if choies == 'a':
            vector_a=list(calculate)
        elif choies == 'b':
            vector_b=list(calculate)

elif choice==6:
    try:
        scalar=np.dot(vector_a,vector_b)
    except ValueError:
        print('OBS!,You miss one of vectors,Try again')
    else:
        print(vector_a,' . ',vector_b,'=',scalar)

elif choice==7:
    try:
        product=np.cross(vector_a,vector_b)
    except ValueError as error:
        print('OBS!,You miss one of vectors,Try again')
    else:
        print(vector_a,'X',vector_b,'=',product)

        print('_____')
        print('Do you want to save the image to file?')
        print('(Answer Y to save, other choices mean no.)')
        choice_image=input('Your choies: ')
        if choice_image=='Y':
            max_value=vector_a+vector_b+list(product)
            list_abs=list(map(lambda x:abs(x),max_value))
            filnam=input('filename: ')
            format='.png'
            filename=filnam + format
            graph_vector(product,'green',vector_a,'red',vector_b,'
blue',filename,max(list_abs))
            print('Do you want to save the result in a or b?')
            print('(Answer a or b to save, other choices mean no.)')
            choies=input('Your choies: ')
            if choies == 'a':
                vector_a=list(product)
            elif choies == 'b':
                vector_b=list(product)

elif choice==8:
    print('The equation for a plane is a * x + b * y + c * z + d =
0')

    plan=the_plan()
    n_vector=plan[:3]

```

```

elif choice==9:
    try:
        graph_plan(plan)
    except ValueError:
        print('OBS!,You miss one of vectors,Try again')

elif choice==10:
    try:
        Project_vector1_on_plane_p(vector_a,n_vector)
        print('projects vectorn
is:',Project_vector1_on_plane_p(vector_a,n_vector))
        graph_project(plan,vector_a,n_vector,Project_vector1_on_pl
ane_p(vector_a,n_vector))
    except ValueError:
        print('OBS!,You miss one of vectors,Try again')
main()
#1. Hur mycket tid trodde du att det skulle ta att l sa den h r
uppgiften? 3-4 dagar
#2. Hur mycket tid har du lagt ned p  att l sa uppgiften? 4-5 dagar

```