

Бюджетное учреждение высшего образования Ханты-Мансийского автономного округа – Югры

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт

Кафедра прикладной математики

Гркиян Мисак Эдикович

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №1

Дисциплина «Алгебра и геометрия»

направление 01.03.02 «Прикладная математика и информатика»

направленность (профиль): «Технологии программирования и анализ данных»

Преподаватель: Шапошникова Ирина Вадимовна

Студент гр. № 601-31

Гркиян Мисак Эдикович

Индивидуальное задание №1

Вариант №7

Задание 1

Даны векторы v , u и точка A . Написать программу на языке Python с использованием библиотек `matplotlib` и `numpy`, которая строит рисунок, изображающий векторы v и u , приложенные в точке A . На рисунке должны быть подписаны оси Ox и Oy , векторы v и u и точка A . Также программа должна вычислить и вывести значение модуля векторного произведения $v \times u$, и изобразить параллелограмм, построенный на векторах v и u . Построенный рисунок должен приблизительно иметь вид, показанный на рис. 3.

7. $v = (-1, -2)$, $u = (-3, 2)$ и т. $A(0, 2)$

Программное решение

```
import numpy as np
import matplotlib.pyplot as plt
import math

if __name__ == "__main__":
    u = np.array([-3, 2])
    A = np.array([0, 2])

def plot_vectors():
    #подписи координатных осей
    plt.xlabel("x", fontsize="xx-large",
               fontstyle="italic", family="serif")
    plt.ylabel("y", fontsize="xx-large",
               fontstyle="italic", family="serif")

    #изображение векторов в виде стрелок
    plt.arrow(A[0], A[1], v[0], v[1], linewidth=2,
              head_width=0.1, head_length=0.1,
              length_includes_head=True)

    plt.arrow(A[0], A[1], u[0], u[1], linewidth=2,
              head_width=0.1, head_length=0.1,
              length_includes_head=True)

    #создание подписей для A, u, v
    dr = np.array([0.05, -0.2])
    dt = np.array([-0.5, 5.1])
    dp = np.array([2.5, -1.3])
```

```

plt.text(A[0]+dr[0], A[1]+dr[1], "A", fontsize="xx-large",
fontstyle="italic", family="serif")
plt.text(v[0]+dt[0], v[1]+dt[1], "v", fontsize="xx-large",
fontstyle="italic", family="serif")
plt.text(u[0]+dp[0], u[1]+dp[1], "u", fontsize="xx-large",
fontstyle="italic", family="serif")

# Вычисляем точку B путем сложения векторов
B = A + v + u #B=(-4; 2)

# Векторы с помощью которых мы будем достраивать до параллелограмма
z = B - v
f = B - u

x1, y1 = zip(z, B)
x2, y2 = zip(f, B)
plt.plot(x1, y1, linewidth=2, color='Blue')
plt.plot(x2, y2, linewidth=2, color='Blue')

plt.show()

plot_vectors()

vA = np.array([A[0]-v[0], A[1]-v[1]])
uA = np.array([A[0]-u[0], A[1]-u[1]])

# Вычисление векторного произведения
vp = np.cross(vA, uA)

# Вычисление модуля векторного произведения
vp_mod = np.linalg.norm(vp)

print("Модуль векторного произведения: ", vp_mod)

```

Иллюстрация решения

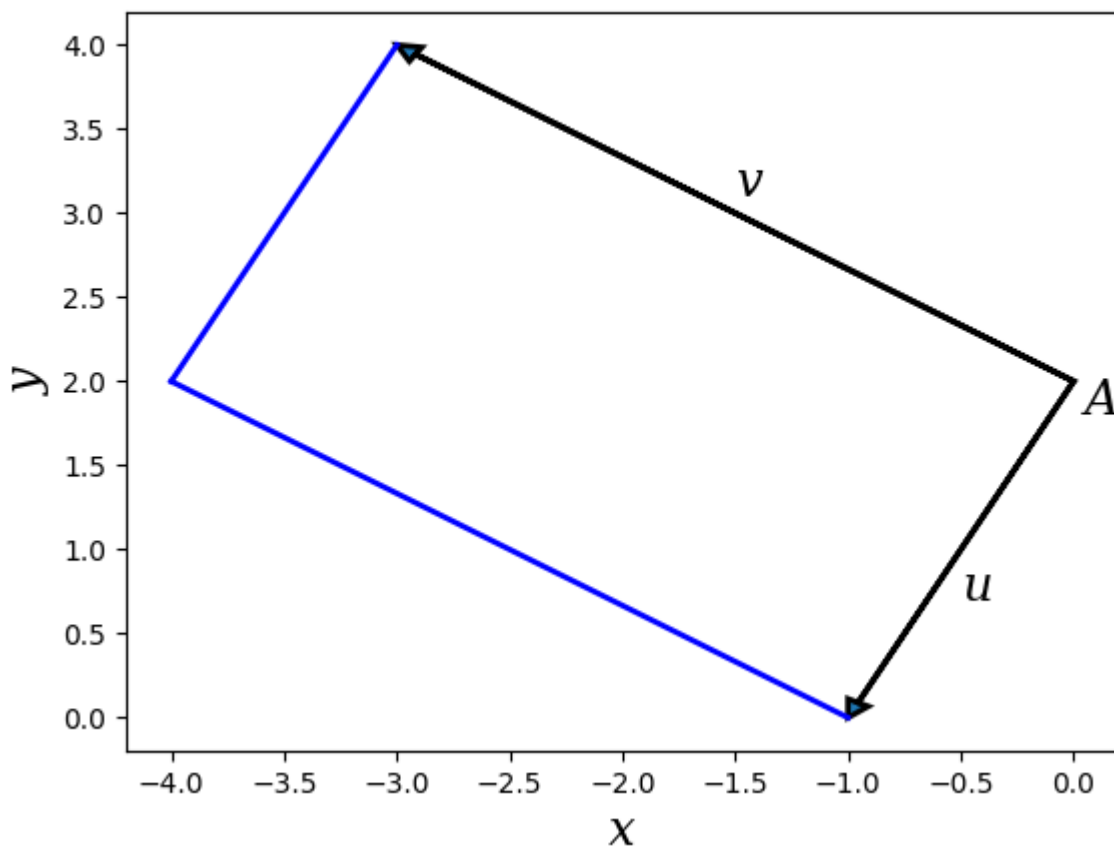


Рис. 1. Иллюстрация решения задачи.

```

● PS C:\Users\acer\Desktop\ALGEBRA> python -u "c:\Users\acer\Desktop\ALGEBRA\lab1\tempCodeRunnerFile.py"
Модуль векторного произведения: 12.0
○ PS C:\Users\acer\Desktop\ALGEBRA>

```

Рис. 2. Вывод программы в терминале.

Задание 2

Даны векторы a , b , c , d . Написать программу на языке Python с использованием библиотек `matplotlib` и `numpy`, которая строит рисунок следующим образом:

1. вектор a приложен в начале координат;
2. вектор b приложен в конце вектора a ;
3. вектор c приложен в конце вектора b ;
4. вектор d приложен в конце вектора c ;
5. вектор $a + b + c + d$ приложен в начале координат.

На рисунке должны быть подписаны оси Ox и Oy , а также все построенные векторы.

7. $a = (1, -1)$, $b = (2, -3)$, $c = (-2, 5)$, $d = (1, 3)$

Программное решение

```
import numpy as np
import matplotlib.pyplot as plt

# Задание векторов
a = np.array([1, -1])
b = np.array([2, -3])
c = np.array([-2, 5])
d = np.array([1, 3])

# Создание фигуры и осей
fig, ax = plt.subplots()

def plot_vectors():
    # Построение векторов
    ax.quiver(0, 0, a[0], a[1], angles='xy', scale_units='xy', scale=1,
color='r', label='a')
    ax.quiver(a[0], a[1], b[0], b[1], angles='xy', scale_units='xy', scale=1,
color='g', label='b')
    ax.quiver(a[0]+b[0], a[1]+b[1], c[0], c[1], angles='xy', scale_units='xy',
scale=1, color='b', label='c')
    ax.quiver(a[0]+b[0]+c[0], a[1]+b[1]+c[1], d[0], d[1], angles='xy',
scale_units='xy', scale=1, color='m', label='d')
    ax.quiver(0, 0, a[0]+b[0]+c[0]+d[0], a[1]+b[1]+c[1]+d[1], angles='xy',
scale_units='xy', scale=1, color='k', label='a+b+c+d')

    # Настройка осей и меток
    ax.set_xlim([-6, 6])
    ax.set_ylim([-6, 6])
    ax.axhline(0, color='black',linewidth=0.5)
    ax.axvline(0, color='black',linewidth=0.5)
    ax.set_xlabel('Ox')
    ax.set_ylabel('Oy')

    # Добавление легенды
    ax.legend()

    # Отображение рисунка
    plt.show()

plot_vectors()
```

Иллюстрация решения

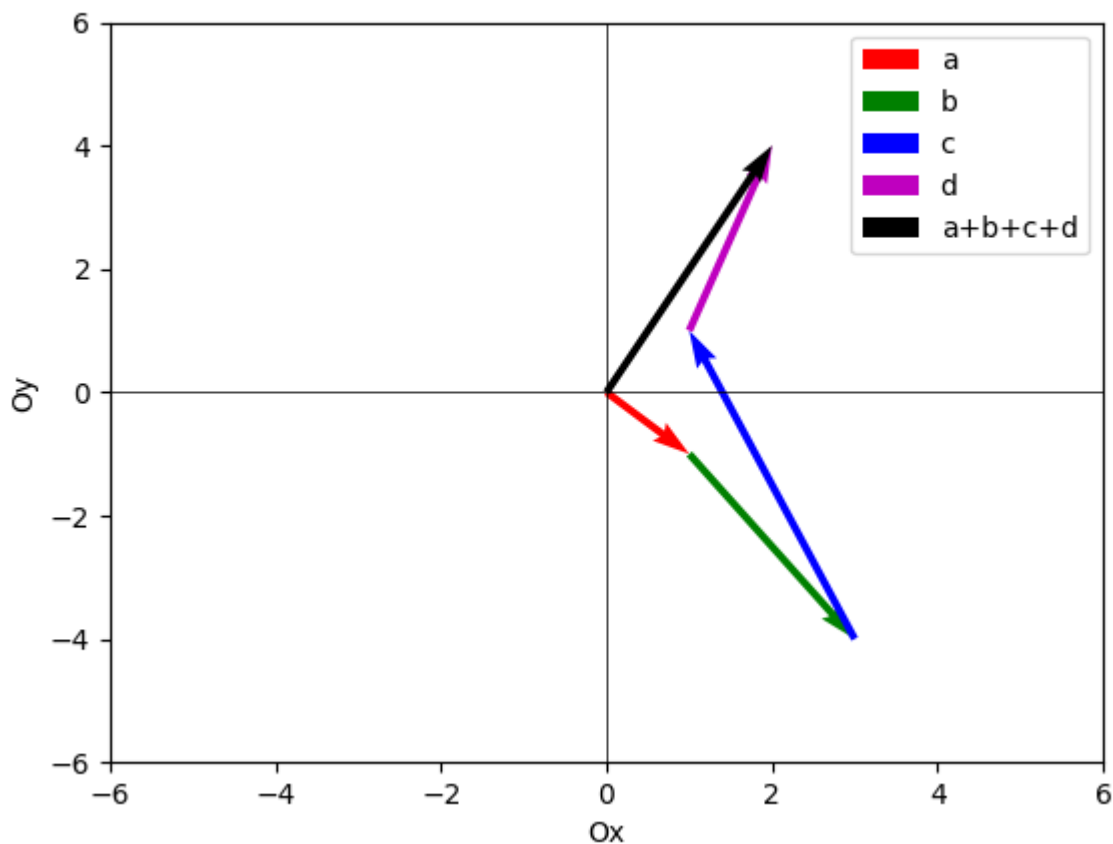


Рис. 1. Иллюстрация решения задачи.

Задание 3

Даны точки A , B , C , D . Написать программу на языке Python с использованием библиотек `matplotlib` и `numpy`, которая строит четырехугольник $ABCD$ и вычисляет его площадь при помощи векторного произведения. На рисунке должны быть подписаны оси Ox и Oy , а также точки A , B , C , D .

7. $A(3,1)$, $B(-2,3)$, $C(-4, 0)$, $D(-1, 3)$

Программное решение

```
import numpy as np
import matplotlib.pyplot as plt

A = np.array([3, 1])
B = np.array([-2, 3])
C = np.array([-4, 0])
D = np.array([-1, 3])

def plot_pic():
    plt.plot([A[0], D[0]], [A[1], D[1]], color='b')
```

```

plt.plot([B[0], C[0]], [B[1], C[1]], color='b')
plt.plot([C[0], A[0]], [C[1], A[1]], color='b')
plt.plot([D[0], B[0]], [D[1], B[1]], color='b')
plt.plot([C[0], D[0]], [C[1], D[1]], color='red', linestyle='--') # Это
потом пригодится для вычисления площади
plt.scatter(A[0], A[1], color='red', label='A=(3, 1)', alpha=1)
plt.scatter(B[0], B[1], color='red', label='B=(-2, 3)', alpha=1)
plt.scatter(C[0], C[1], color='red', label='C=(-4, 0)', alpha=1)
plt.scatter(D[0], D[1], color='red', label='D=(-1, 3)', alpha=1)

#создание подписей для A, u, v
dA = np.array([0.0, 0.15])
dB = np.array([-0.6, -0.1])
dC = np.array([-0.3, 0.1])
dD = np.array([0.4, -0.1])

plt.text(A[0]+dA[0], A[1]+dA[1], "A", fontsize="xx-large",
fontstyle="italic", family="serif")
plt.text(B[0]+dB[0], B[1]+dB[1], "B", fontsize="xx-large",
fontstyle="italic", family="serif")
plt.text(C[0]+dC[0], C[1]+dC[1], "C", fontsize="xx-large",
fontstyle="italic", family="serif")
plt.text(D[0]+dD[0], D[1]+dD[1], "D", fontsize="xx-large",
fontstyle="italic", family="serif")

plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.grid(True, linewidth=0.5, linestyle='dotted')

plt.show()

plot_pic()

# Вычисление площади четырехугольника ABCD
# Для начала вычислим площади треугольников, которые образовались из-за диагонали
CD
def calculate_bdc():
    square_bdc = 0.5 * abs(((D[0]-C[0]) * (B[1]-C[1]) - (B[0]-C[0]) * (D[1]-
C[1])))
    print(f"Площадь треугольника BDC = {square_bdc}")
    return square_bdc

square1 = calculate_bdc()

def calculate_adc():
    square_adc = 0.5 * abs(((D[0]-C[0]) * (A[1]-C[1]) - (A[0]-C[0]) * (D[1]-
C[1])))
    print(f"Площадь треугольника ADC = {square_adc}")
    return square_adc

square2 = calculate_adc()

# Вычислим площадь четырехугольника ABCD, просуммировав площади треугольников

```

```
sum = square1 + square2  
print(f"Площадь четырехугольника ABCD = {sum}")
```

Иллюстрация решения

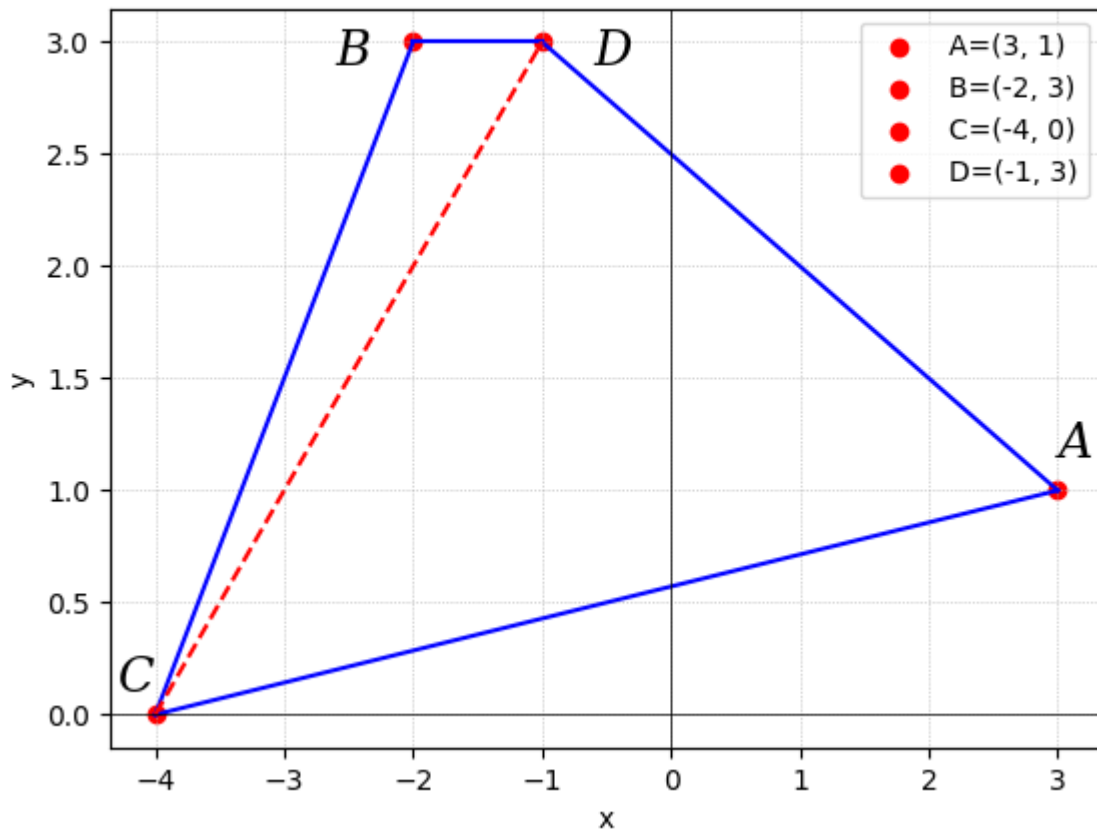


Рис. 1. Иллюстрация решения задачи.

```
PS C:\Users\acer\Desktop\ALGEBRA> python -u "c:\Users\acer\Desktop\ALGEBRA\lab1\app3.py"  
Площадь треугольника BDC = 1.5  
Площадь треугольника ADC = 9.0  
Площадь четырехугольника ABCD = 10.5  
PS C:\Users\acer\Desktop\ALGEBRA>
```

Рис. 2. Вывод программы в терминале.