**Задание 2**

1. **import** numpy **as** np
2. **from** numpy **import** exp,sqrt
3. least\_route = []
5. X = list(map(**lambda** x: int(x), list(filter(None, input("Введите координаты X точек на плате  ").split()))))
6. Y = list(map(**lambda** x: int(x), list(filter(None, input("Введите координаты Y точек на плате  ").split()))))
7. **if** (len(X)!= len(Y)):
8. exit("Количество координат не равны друг другу")
9. n = len(X)
11. route\_total\_distances=[]
12. routes=[]

15. **def** distancew(one, two):
16. **return** sqrt((X[least\_route[one]] - X[least\_route[two]]) \*\* 2 + (Y[least\_route[one]] - Y[least\_route[two]]) \*\* 2)

19. **for** ib **in** np.arange(0,n,1):
20. M = np.zeros([n,n])
21. **for** i **in** np.arange(0,n,1):
22. **for** j **in** np.arange(0,n,1):
23. **if** i!=j:
24. M[i,j]=sqrt((X[i]-X[j])\*\*2+(Y[i]-Y[j])\*\*2)
25. **else**:
26. M[i,j]=float('inf')


30. least\_route=[]
31. least\_route.append(ib)
32. **for** i **in** np.arange(1,n,1):
33. s=[]
34. **for** j **in** np.arange(0,n,1):
35. s.append(M[least\_route[i - 1], j])
36. least\_route.append(s.index(min(s)))
37. **for** j **in** np.arange(0,i,1):
38. M[least\_route[i], least\_route[j]]=float('inf')
40. S= sum([distancew(i, i+1) **for** i **in** np.arange(0, n - 1, 1)]) + distancew(0, n-1)
41. route\_total\_distances.append(S)
42. routes.append(least\_route)
44. least\_total\_distance=min(route\_total\_distances)
45. least\_route=routes[route\_total\_distances.index(least\_total\_distance)]
47. **print**("Кратчайший маршрут должен пролегать через точки в порядке: ")
48. **for** i **in** least\_route:
49. **print**((X[i], Y[i]))