

```
1  # 12-1
2  print("stack")
3  n = 0
4  stack = []
5  while True:
6      print("-----")
7      print("stack operation")
8      print("option 1: Push")
9      print("option 2: Pop")
10     print("option 3: Display")
11     print("option 4: Quit")
12     print("-----")
13
14     choice = eval(input("Input your choice: "))
15     if choice == 1:
16         key = eval(input("Input your key: "))
17         stack.append(key)
18         n += 1
19     elif choice == 2:
20         if n > 0:
21             key = stack.pop()
22             print("Pop key: ", key)
23             n -= 1
24         else:
25             print("Stack is empty")
26     elif choice == 3:
27         print("Stack: ", stack)
28     else:
29         break
```

```
31 # 12-2
32 print("Queue")
33 n = 0
34 queue = []
35 while True:
36     print("-----")
37     print("Queue operation")
38     print("option 1: Enqueue")
39     print("option 2: dequeue")
40     print("option 3: Display")
41     print("option 4: Quit")
42     print("-----")
43
44     choice = eval(input("Input your choice: "))
45     if choice == 1:
46         key = eval(input("Input your key: "))
47         queue.append(key)
48         n += 1
49     elif choice == 2:
50         if n > 0:
51             key = queue.pop(0)
52             print("Pop key: ", key)
53             n -= 1
54         else:
55             print("Queue is empty")
56     elif choice == 3:
57         print("Queue: ", queue)
58     else:
59         break
```

```

61 # 12-3
62 a = [1, 2, 3, 4]
63 b = [2, 4, 3, 1]
64
65 c = [0 for i in range(4)]
66 for i in range(4):
67     c[i] = a[i] + b[i]
68 print("向量加法")
69 print(c)
70
71 scalar = 3
72 for i in range(4):
73     c[i] = scalar * a[i]
74 print("向量乘法")
75 print(c)
76
77 # 12-4
78 def Matrix_Add(A, B): 1 usage new *
79     n = len(A)
80     c = [[0 for j in range(n)] for i in range(n)]
81     for i in range(n):
82         for j in range(n):
83             c[i][j] = A[i][j] + B[i][j]
84     return c
85 def Matrix_Multiply(A, B): 1 usage new *
86     n = len(A)
87     c = [[0 for j in range(n)] for i in range(n)]
88     for i in range(n):
89         for j in range(n):

```

```

78 def Matrix_Add(A, B): 1 usage new *
79     c = [[0 for j in range(n)] for i in range(n)]
80
81     for i in range(n):
82         for j in range(n):
83             c[i][j] = A[i][j] + B[i][j]
84     return c
85
86 def Matrix_Multiply(A, B): 1 usage new *
87     n = len(A)
88     c = [[0 for j in range(n)] for i in range(n)]
89     for i in range(n):
90         for j in range(n):
91             for k in range(n):
92                 c[i][j] = c[i][j] + A[i][k] * B[k][j]
93
94     return c
95
96 A = [[1, 2], [3, 4]]
97 B = [[2, 4], [3, 1]]
98
99 c = Matrix_Add(A, B)
100 print("矩陣乘法")
101 print(c)
102
103 c = Matrix_Multiply(A, B)
104 print("矩陣乘法")
105 print(c)

```

```
/usr/local/bin/python3.12 /Users/pengyenjia/Desktop/運算思維與程式
```

```
stack
```

```
-----
```

```
stack operation
```

```
option 1: Push
```

```
option 2: Pop
```

```
option 3: Display
```

```
option 4: Quit
```

```
-----
```

```
Input your choice: 1
```

```
Input your key: 1
```

```
-----
```

```
stack operation
```

```
option 1: Push
```

```
option 2: Pop
```

```
option 3: Display
```

```
option 4: Quit
```

```
-----
```

```
Input your choice: 1
```

```
Input your key: 2
```

```
-----
```

```
stack operation
```

```
option 1: Push
```

```
option 2: Pop
```

```
option 3: Display
```

```
option 4: Quit
```

```
-----
```

```
Input your choice: 1
```

```
Input your key: 3
```

```
-----
```

```
stack operation
```

```
option 1: Push
```

```
option 2: Pop
```

```
-----  
Input your choice: 2  
Pop key: 3  
-----
```

```
stack operation  
option 1: Push  
option 2: Pop  
option 3: Display  
option 4: Quit  
-----
```

```
Input your choice: 3  
Stack: [1, 2]  
-----
```

```
stack operation  
option 1: Push  
option 2: Pop  
option 3: Display  
option 4: Quit  
-----
```

```
Input your choice: 4  
Queue  
-----
```

```
Queue operation  
option 1: Enqueue  
option 2: dequeue  
option 3: Display  
option 4: Quit  
-----
```

```
Input your choice: 1  
Input your key: 1  
-----
```

```
Queue operation  
option 1: Enqueue
```

Input your choice: 1

Input your key: 1

Queue operation

option 1: Enqueue

option 2: dequeue

option 3: Display

option 4: Quit

Input your choice: 1

Input your key: 2

Queue operation

option 1: Enqueue

option 2: dequeue

option 3: Display

option 4: Quit

Input your choice: 1

Input your key: 3

Queue operation

option 1: Enqueue

option 2: dequeue

option 3: Display

option 4: Quit

Input your choice: 2

Pop key: 1

Queue operation

option 1: Enqueue

option 2: dequeue

```
option 3: Display
option 4: Quit
-----
Input your choice: 2
Pop key: 1
-----
```

```
Queue operation
option 1: Enqueue
option 2: dequeue
option 3: Display
option 4: Quit
-----
Input your choice: 3
Queue:  [2, 3]
-----
```

```
Queue operation
option 1: Enqueue
option 2: dequeue
option 3: Display
option 4: Quit
-----
Input your choice: 4
```

```
向量加法
[3, 6, 6, 5]
向量乘法
[3, 6, 9, 12]
矩陣乘法
[[3, 6], [6, 5]]
矩陣乘法
[[8, 6], [18, 16]]
```

```
Process finished with exit code 0
```