

Python Code

```
1 """
2 =====
3 NUMPY / MATRIX OPERATIONS (IMAGE 4 & 5)
4 =====
5 Matrix A =
6 [[2, 4, 6],
7  [8, 0, 1],
8  [3, 5, 7]]
9 Functional updates for Matrix A
10 1) Add row at last
11 2) Add column at last
12 3) Add row at pos1
13 4) Add column at pos2
14 """
15 A = [[2, 4, 6], [8, 0, 1], [3, 5, 7]]
16 print("Matrix A = ")
17 for row in A: print(row)
18
19 xlist_A = [
20     (A.append([10, 11, 12]), A.copy())[1],
21     (A[0].append(13), A[1].append(14), A[2].append(15), A.copy())[3],
22     (A.insert(1, [16, 17, 18, 19]), A.copy())[1],
23     (A[0].insert(2,19), A[1].insert(2,20), A[2].insert(2,21), A.copy())[3]
24 ]
25
26 for i, x in enumerate(xlist_A):
27     print(f"{i+1}. {x}")
28
29 print("=====")
30 ######
31
32 """
33 -----
34 Matrix B (4x4) =
35 [[1, 2, 3, 4],
36  [5, 6, 7, 8],
37  [9, 10, 11, 12],
38  [13, 14, 15, 16]]
39 1) 2nd row
40 2) 3rd column
41 3) Rows 3 and 4
42 4) First 2 columns
43 5) Sub-matrix [10..16]
44 6) Elements [5,8], [13,16]
45 7) Diagonal
46 8) Anti-diagonal
47 """
```

```

48 B = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
49
50 xlist_B = [
51     B[1],
52     [row[2] for row in B],
53     B[2:],
54     [row[:2] for row in B],
55     [row[1:] for row in B[1:]],
56     [[B[1][0], B[1][3]], [B[3][0], B[3][3]]],
57     [B[i][i] for i in range(len(B))],
58     [B[i][len(B)-1-i] for i in range(len(B))]
59 ]
60
61 for i, x in enumerate(xlist_B):
62     print(f"{i+1}. {x}")
63
64 print("====")
65 #####
66
67 """
68 =====
69 MATRIX SLICING & MANIPULATION (IMAGE 6)
70 =====
71 Matrix C (3x3)
72 Add col [0,0,0] last
73 Add col [5,5,5] pos 1
74 Add row [1..5] last
75 Add row [6..10] pos 2
76 Add col [10..50] last
77 """
78 C = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
79
80 xlist_C = [
81     ([C[i].append(0) for i in range(3)], C.copy())[1],
82     ([C[i].insert(1, 5) for i in range(3)], C.copy())[1],
83     (C.append([1, 2, 3, 4, 5]), C.copy())[1],
84     (C.insert(2, [6, 7, 8, 9, 10]), C.copy())[1],
85     ([C[i].append(val) for i, val in enumerate([10,20,30,40,50])], C.copy())[1]
86 ]
87
88 for i, x in enumerate(xlist_C):
89     print(f"{i+1}. {x}")
90
91 print("====")
92 #####
93
94 """
95 Matrix D (5x5)
96 Slice specific rows
97 Step slicing columns

```

```

98 Sub-matrix [6,7,8], [16,17,18]
99 Diagonal
100 Anti-diagonal
101 Selective elements
102 Elements 10, 15, 20
103 Elements [8,9], [18,19]
104 Last row
105 Sub-matrix[2,4]..
106 """
107 D = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15], [16, 17, 18, 19, 20], [21, 22,
108 23, 24, 25]]
109 xlist_D = [
110     [D[0], D[2], D[4]],
111     [[row[0], row[2], row[4]] for row in D],
112     [D[1][:3], D[3][:3]],
113     [D[i][i] for i in range(5)],
114     [D[i][4-i] for i in range(5)],
115     [[D[i][j] for j in [0, 2, 4]] for i in [0, 2, 4]],
116     [D[1][4], D[2][4], D[3][4]],
117     [D[1][2:4], D[3][2:4]],
118     D[4],
119     [D[i][1:4:2] for i in [0, 2, 4]]
120 ]
121
122 for i, x in enumerate(xlist_D):
123     print(f"{i+1}. {x}")
124
125 print("====")
126 #####"
127 """
128
129 Matrix E (3x3)
130 a) Replace 60 with 6
131 b) Rows 1 and 2
132 c) Sub-matrix [50,6..]
133 d) Diagonal
134 e)Anti-diagonal
135 f) Middle column
136 g) Middle column
137 h) 40 and 6
138 i) 10 and 70
139 i) [2030], [80,90]
140 j) [40,6], [70,90]
141 """
142 E = [[10, 20, 30], [40, 50, 60], [70, 80, 90]]
143
144 xlist_E = [
145     (E[1].__setitem__(2, 6), E.copy())[1],
146     E[:2],

```

```

147 [row[1:] for row in E[1:]],
148 [E[i][i] for i in range(3)],
149 [E[i][2-i] for i in range(3)],
150 [row[1] for row in E],
151 [E[1][0], E[1][2]],
152 [E[0][0], E[2][0]],
153 [E[0][1:], E[2][1:]],
154 [[E[1][0], E[1][2]], [E[2][0], E[2][2]]]
155 ]
156
157 for i, x in enumerate(xlist_E):
158     print(f"{i+1}. {x}")
159
160 print("====")
161 #####
162 """
163 """
164 Matrix F (4x5)
165 1) Element 12
166 2) Columns [9,3,2]..
167 3) Elements 9 and 0
168 4) Sub-matrices
169 5) Elements [11,4,5]..
170 6) Add row 0s
171 7) [11,0,4,6,5] (approx)
172 8) Reverse first row
173 9) Rows 2 and 3
174 0) Multiple sub 1-slices
175 """
176 F = [[11, 0, 4, 6, 5], [1, 9, 3, 2, 1], [7, 0, 4, 9, 8], [3, 7, 12, 15, 0]]
177
178 xlist_F = [
179     F[3][2],
180     [[F[1][1], F[1][2], F[1][3]], [F[2][1], F[2][2], F[2][3]]],
181     [F[1][1], F[2][1]],
182     [F[i][:3] for i in [1, 2, 3]],
183     [[F[0][0], F[0][2], F[0][4]], [F[3][0], F[3][2], F[3][4]]],
184     (F.append([0, 0, 0, 0, 0]), F.copy())[1],
185     F[0],
186     F[0][::-1],
187     F[1:3],
188     [F[0][2:4], F[1][2:4], F[2][2:4], F[3][2:4]]
189 ]
190
191 for i, x in enumerate(xlist_F):
192     print(f"{i+1}. {x}")
193
194 print("====")
195 #####

```

```
PS C:\Internship\Kakunje\day4> python task.py
Matrix A =
[2, 4, 6]
[8, 0, 1]
[3, 5, 7]
1. [[2, 4, 19, 6, 13], [8, 0, 21, 1, 14], [3, 5, 7, 15], [10, 11, 12]]
2. [[2, 4, 19, 6, 13], [8, 0, 21, 1, 14], [3, 5, 7, 15], [10, 11, 12]]
3. [[2, 4, 19, 6, 13], [16, 17, 20, 18, 19], [8, 0, 21, 1, 14], [3, 5, 7, 15], [10, 11, 12]]
4. [[2, 4, 19, 6, 13], [16, 17, 20, 18, 19], [8, 0, 21, 1, 14], [3, 5, 7, 15], [10, 11, 12]]
=====
1. [5, 6, 7, 8]
2. [3, 7, 11, 15]
3. [[9, 10, 11, 12], [13, 14, 15, 16]]
4. [[1, 2], [5, 6], [9, 10], [13, 14]]
5. [[6, 7, 8], [10, 11, 12], [14, 15, 16]]
6. [[5, 8], [13, 16]]
7. [1, 6, 11, 16]
8. [4, 7, 10, 13]
=====
1. [[1, 5, 1, 1, 0, 10], [1, 5, 1, 1, 0, 20], [1, 5, 1, 1, 0, 40]]
2. [[1, 5, 1, 1, 0, 10], [1, 5, 1, 1, 0, 20], [1, 5, 1, 1, 0, 40]]
3. [[1, 5, 1, 1, 0, 10], [1, 5, 1, 1, 0, 20], [1, 5, 1, 1, 0, 40], [1, 2, 3, 4, 5, 50]]
4. [[1, 5, 1, 1, 0, 10], [1, 5, 1, 1, 0, 20], [6, 7, 8, 9, 10, 30], [1, 5, 1, 1, 0, 40], [1, 2, 3, 4, 5, 50]]
5. [[1, 5, 1, 1, 0, 10], [1, 5, 1, 1, 0, 20], [6, 7, 8, 9, 10, 30], [1, 5, 1, 1, 0, 40], [1, 2, 3, 4, 5, 50]]
=====
1. [[1, 2, 3, 4, 5], [11, 12, 13, 14, 15], [21, 22, 23, 24, 25]]
2. [[1, 3, 5], [6, 8, 10], [11, 13, 15], [16, 18, 20], [21, 23, 25]]
3. [[6, 7, 8], [16, 17, 18]]
4. [1, 7, 13, 19, 25]
5. [5, 9, 13, 17, 21]
6. [[1, 3, 5], [11, 13, 15], [21, 23, 25]]
7. [10, 15, 20]
8. [[8, 9], [18, 19]]
9. [21, 22, 23, 24, 25]
10. [[2, 4], [12, 14], [22, 24]]
=====
1. [[10, 20, 30], [40, 50, 6], [70, 80, 90]]
2. [[10, 20, 30], [40, 50, 6]]
3. [[50, 6], [80, 90]]
4. [10, 50, 90]
5. [30, 50, 70]
6. [20, 50, 80]
7. [40, 6]
8. [10, 70]
9. [[20, 30], [80, 90]]
10. [[40, 6], [70, 90]]
=====
1. 12
2. [[9, 3, 2], [0, 4, 9]]
3. [9, 0]
4. [[1, 9, 3], [7, 0, 4], [3, 7, 12]]
5. [[11, 4, 5], [3, 12, 0]]
6. [[11, 0, 4, 6, 5], [1, 9, 3, 2, 1], [7, 0, 4, 9, 8], [3, 7, 12, 15, 0], [0, 0, 0, 0, 0]]
7. [11, 0, 4, 6, 5]
8. [5, 6, 4, 0, 11]
9. [[1, 9, 3, 2, 1], [7, 0, 4, 9, 8]]
10. [[4, 6], [3, 2], [4, 9], [12, 15]]
```