

# Assignment 1

2347139

November 7, 2023

Implement Matrix manipulation . Consider the 2D representation for your chosen domain. Perform all data structure operations (insertion, Deletion, linear search) using 2D arrays for any chosen logical data of your domain. Implement any two matrix operations.

## 2D array operations and matrix

```
1  #include <stdio.h>
2  #include <string.h>
3  #include "insert.h"
4
5  #define MAX_HONEYPOTS 100
6
7  struct Honeypot
8  {
9      int honeypotId;
10     char name[50];
11     float port;
12 };
13
14 struct Honeypot Honeypots[MAX_HONEYPOTS];
15 int numHoneypots = 0;
16
17 void insertHoneypot()
18 {
19     if (numHoneypots < MAX_HONEYPOTS)
20     {
21         struct Honeypot newHoneypot;
22         printf("Enter Honeypot ID: ");
23         scanf("%d", &newHoneypot.honeypotId);
24         printf("Enter Honeypot Name: ");
25         scanf("%s", newHoneypot.name);
26         printf("Enter Honeypot port: ");
27         scanf("%f", &newHoneypot.port);
28
29         Honeypots[numHoneypots] = newHoneypot;
30         numHoneypots++;
31         printf("Honeypot inserted successfully!\n");
```

```

32     }
33     else
34     {
35         printf("Maximum number of Honeypots reached.\n");
36     }
37 }
38
39 void deleteHoneypot(int honeypotId)
40 {
41     int i, j;
42     for (i = 0; i < numHoneypots; i++)
43     {
44         if (Honeypots[i].honeypotId == honeypotId)
45         {
46             for (j = i; j < numHoneypots - 1; j++)
47             {
48                 Honeypots[j] = Honeypots[j + 1];
49             }
50             numHoneypots--;
51             printf("Honeypot with process ID %d deleted successfully!\n"
52                 , honeypotId);
53             return;
54         }
55     }
56     printf("Honeypot with ID %d not found.\n", honeypotId);
57 }
58
59 int linearSearchHoneypot(int honeypotId)
60 {
61     int i;
62     for (i = 0; i < numHoneypots; i++)
63     {
64         if (Honeypots[i].honeypotId == honeypotId)
65         {
66             return i;
67         }
68     }
69     return -1; // Honeypot not found
70 }
71
72 void displayHoneypots()
73 {
74     int i;
75     printf("Honeypot List:\n");
76     printf("ID\tName\tPORT\n");
77     for (i = 0; i < numHoneypots; i++)
78     {
79         printf("%d\t%s\t%.2f\n", Honeypots[i].honeypotId, Honeypots[i].
80             name, Honeypots[i].port);
81     }
82 }

```

```

82 void addMatrix(float matrix1[][2], float matrix2[][2], float result
    [][][2])
83 {
84     int i, j;
85     for (i = 0; i < 2; i++)
86     {
87         for (j = 0; j < 2; j++)
88         {
89             result[i][j] = matrix1[i][j] + matrix2[i][j];
90         }
91     }
92 }
93
94 void subtractMatrix(float matrix1[][2], float matrix2[][2], float result
    [][][2])
95 {
96     int i, j;
97     for (i = 0; i < 2; i++)
98     {
99         for (j = 0; j < 2; j++)
100        {
101            result[i][j] = matrix1[i][j] - matrix2[i][j];
102        }
103    }
104 }
105
106 int main()
107 {
108     int i, j;
109     int choice;
110     int honeypotId;
111     int searchResult;
112     float matrix1[2][2] = {{1.0, 2.0}, {3.0, 4.0}};
113     float matrix2[2][2] = {{2.0, 1.0}, {4.0, 3.0}};
114     float resultMatrix[2][2];
115
116     do
117     {
118         printf("\nHoneypot Network Menu:\n");
119         printf("1. Insert honeypot to the network\n");
120         printf("2. Delete honeypot to the network\n");
121         printf("3. Search honeypot to the network\n");
122         printf("4. Display honeypot on the network\n");
123         printf("5. Add Matrices\n");
124         printf("6. Subtract Matrices\n");
125         printf("7. Exit\n");
126         printf("Enter your choice: ");
127         scanf("%d", &choice);
128
129         switch (choice)
130         {
131             case 1:

```

```

132         insertHoneypot();
133         break;
134     case 2:
135         printf("Enter Honeypot ID to delete: ");
136         scanf("%d", &honeypotId);
137         deleteHoneypot(honeypotId);
138         break;
139     case 3:
140         printf("Enter Honeypot ID to search: ");
141         scanf("%d", &honeypotId);
142         searchResult = linearSearchHoneypot(honeypotId);
143         if (searchResult != -1)
144         {
145             printf("Honeypot found at index %d\n", searchResult);
146         }
147         else
148         {
149             printf("Honeypot not found.\n");
150         }
151         break;
152     case 4:
153         displayHoneypots();
154         break;
155     case 5:
156         addMatrix(matrix1, matrix2, resultMatrix);
157         printf("Result of Matrix Addition:\n");
158         for (i = 0; i < 2; i++)
159         {
160             for (j = 0; j < 2; j++)
161             {
162                 printf("%.2f\t", resultMatrix[i][j]);
163             }
164             printf("\n");
165         }
166         break;
167     case 6:
168         subtractMatrix(matrix1, matrix2, resultMatrix);
169         printf("Result of Matrix Subtraction:\n");
170         for (i = 0; i < 2; i++)
171         {
172             for (j = 0; j < 2; j++)
173             {
174                 printf("%.2f\t", resultMatrix[i][j]);
175             }
176             printf("\n");
177         }
178         break;
179     case 7:
180         printf("Exiting program. No Honeypots found!\n");
181         break;
182     default:
183         printf("Invalid choice. Please try again.\n");

```

```
184         }
185     } while (choice != 7);
186
187     return 0;
188 }
```