

CSE 358 LAB 3

Arnav Jain - 220002018

Question 1

Code:

```
q1.c x
C q1.c > LIS(int [], int)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void generateRandomArray(int arr[], int size) {
6      for (int i = 0; i < size; i++) {
7          arr[i] = rand() % 100;
8      }
9  }
10
11
12  int LIS(int arr[], int size) {
13
14      int lis[size];
15      for (int i = 0; i < size; i++) {
16          lis[i] = 1;
17      }
18
19      for (int i = 1; i < size; i++) {
20          for (int j = 0; j < i; j++) {
21              if (arr[i] > arr[j] && lis[i] < lis[j] + 1) {
22                  lis[i] = lis[j] + 1;
23              }
24          }
25      }
26
27      int maxLIS = 0;
28      for (int i = 0; i < size; i++) {
29          if (lis[i] > maxLIS) {
30              maxLIS = lis[i];
31          }
32      }
33
34      return maxLIS;
35  }
```

```
C q1.c x
C q1.c > LIS(int [], int)
12 int LIS(int arr[], int size) {
36
37 int main() {
38
39     int size;
40     printf("Enter the size of the array: ");
41     scanf("%d", &size);
42
43     int arr[size];
44     generateRandomArray(arr, size);
45
46     printf("Generated array: ");
47     for (int i = 0; i < size; i++) {
48         printf("%d ", arr[i]);
49     }
50     printf("\n");
51
52     int result = LIS(arr, size);
53     printf("Length of Longest Increasing Subsequence (LIS): %d\n", result);
54
55     return 0;
56 }
```

Result:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ ./q1
Enter the size of the array: 20
Generated array: 83 86 77 15 93 35 86 92 49 21 62 27 90 59 63 26 40 26 72 36
Length of Longest Increasing Subsequence (LIS): 6
○ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$
```

Question 2

Code:

```
C q2.c x
C q2.c > findSmallestMissingPositive(int [], int)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5
6  void generateRandomArray(int arr[], int size) {
7      for (int i = 0; i < size; i++) {
8          arr[i] = (rand() % 100) ;
9      }
10 }
11
12
13 int findSmallestMissingPositive(int arr[], int size) {
14     for (int i = 0; i < size; i++) {
15         while (arr[i] > 0 && arr[i] <= size && arr[arr[i] - 1] != arr[i]) {
16             int temp = arr[i];
17             arr[i] = arr[arr[i] - 1];
18             arr[temp - 1] = temp;
19         }
20     }
21
22     for (int i = 0; i < size; i++) {
23         if (arr[i] != i + 1) {
24             return i + 1;
25         }
26     }
27
28     return size + 1;
29 }
30
31
```

```

31
32 int main() {
33
34     int size;
35     printf("Enter the size of the array: ");
36     scanf("%d", &size);
37
38     int arr[size];
39     generateRandomArray(arr, size);
40
41     printf("Generated array: ");
42     for (int i = 0; i < size; i++) {
43         printf("%d ", arr[i]);
44     }
45     printf("\n");
46
47     int result = findSmallestMissingPositive(arr, size);
48     printf("Smallest missing positive integer (MEX): %d\n", result);
49
50     return 0;
51 }
52

```

Result:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER

● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ ./q2
Enter the size of the array: 20
Generated array: 83 86 77 15 93 35 86 92 49 21 62 27 90 59 63 26 40 26 72 36
Smallest missing positive integer (MEX): 1
○ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ █

```

Question 3

Code:

```
C q3.c x
C q3.c > ...
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4
5
6  void normalizeString(char* str, char* normalized) {
7      int j = 0;
8      for (int i = 0; str[i] != '\0'; i++) {
9          if (isalnum(str[i])) {
10             normalized[j++] = tolower(str[i]);
11         }
12     }
13     normalized[j] = '\0';
14 }
15
16 int isPalindrome(char* str) {
17     int len = strlen(str);
18     for (int i = 0; i < len / 2; i++) {
19         if (str[i] != str[len - 1 - i]) {
20             return 0;
21         }
22     }
23     return 1;
24 }
25
26
27 int isRotatedPalindrome(char* str) {
28     char normalized[1000];
29     normalizeString(str, normalized);
30
31     int len = strlen(normalized);
32
33     for (int i = 0; i < len; i++) {
34         char rotated[1000];
35
36         for (int j = 0; j < len; j++) {
37             rotated[j] = normalized[(i + j) % len];
38         }
39         rotated[len] = '\0';
40
41         if (isPalindrome(rotated)) {
42             return 1;
43         }
44     }
45     return 0;
46 }
```

```

47
48 int main() {
49
50     char str[1000];
51
52     printf("Enter a string: ");
53     fgets(str, sizeof(str), stdin);
54
55     str[strcspn(str, "\n")] = '\0';
56
57     if (isRotatedPalindrome(str)) {
58         printf("The string is a rotated palindrome.\n");
59     } else {
60         printf("The string is not a rotated palindrome.\n");
61     }
62
63     return 0;
64 }
65

```

Result:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER

● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ ./q3
Enter a string: aab
The string is a rotated palindrome.
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ ./q3
Enter a string: abc
The string is not a rotated palindrome.
○ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiler/LAB 3$ █

```

GitHub

<https://github.com/arnavjain2710/Compiler-Techniques/tree/main/LAB%203>