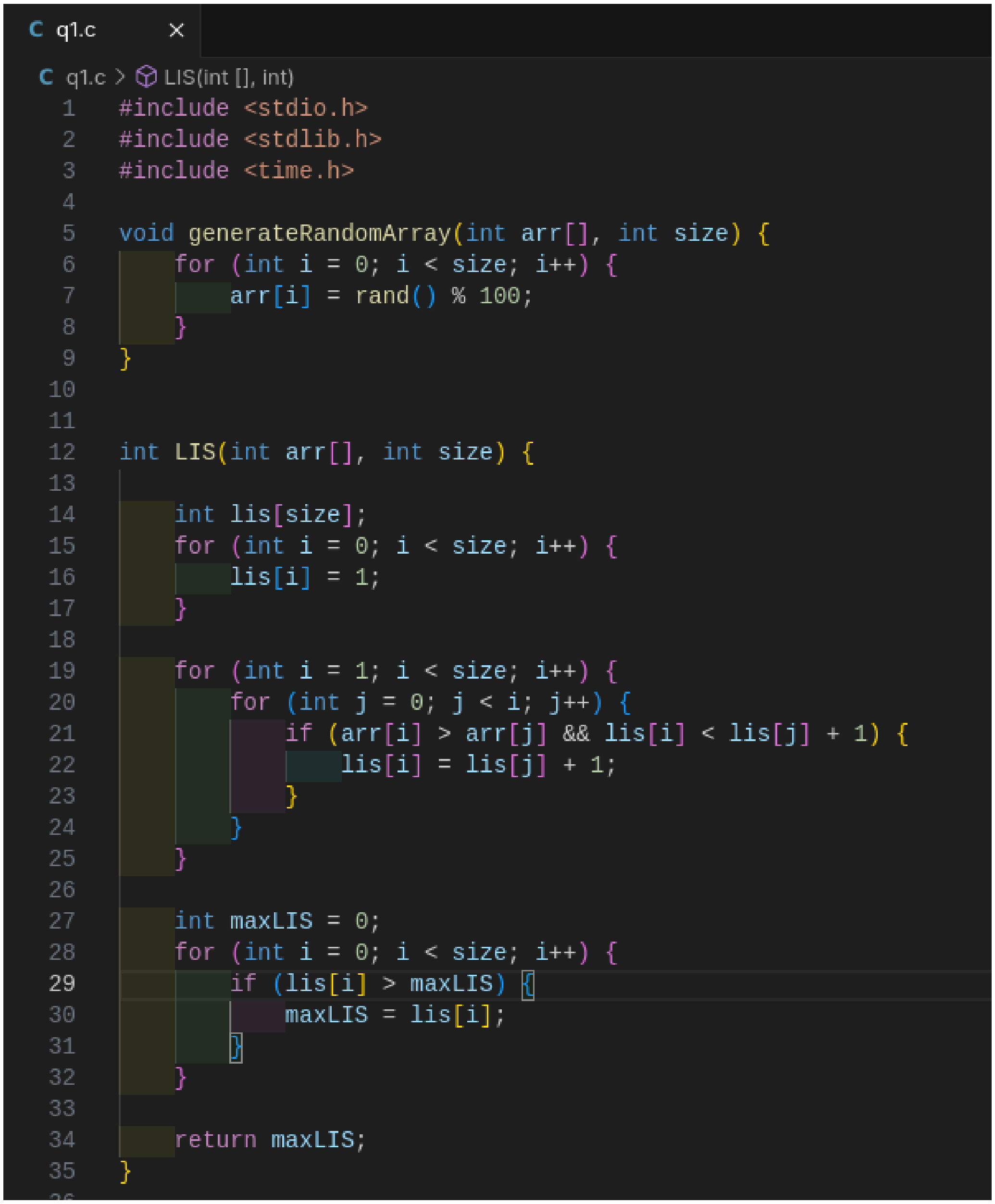
CSE 358 LAB 3

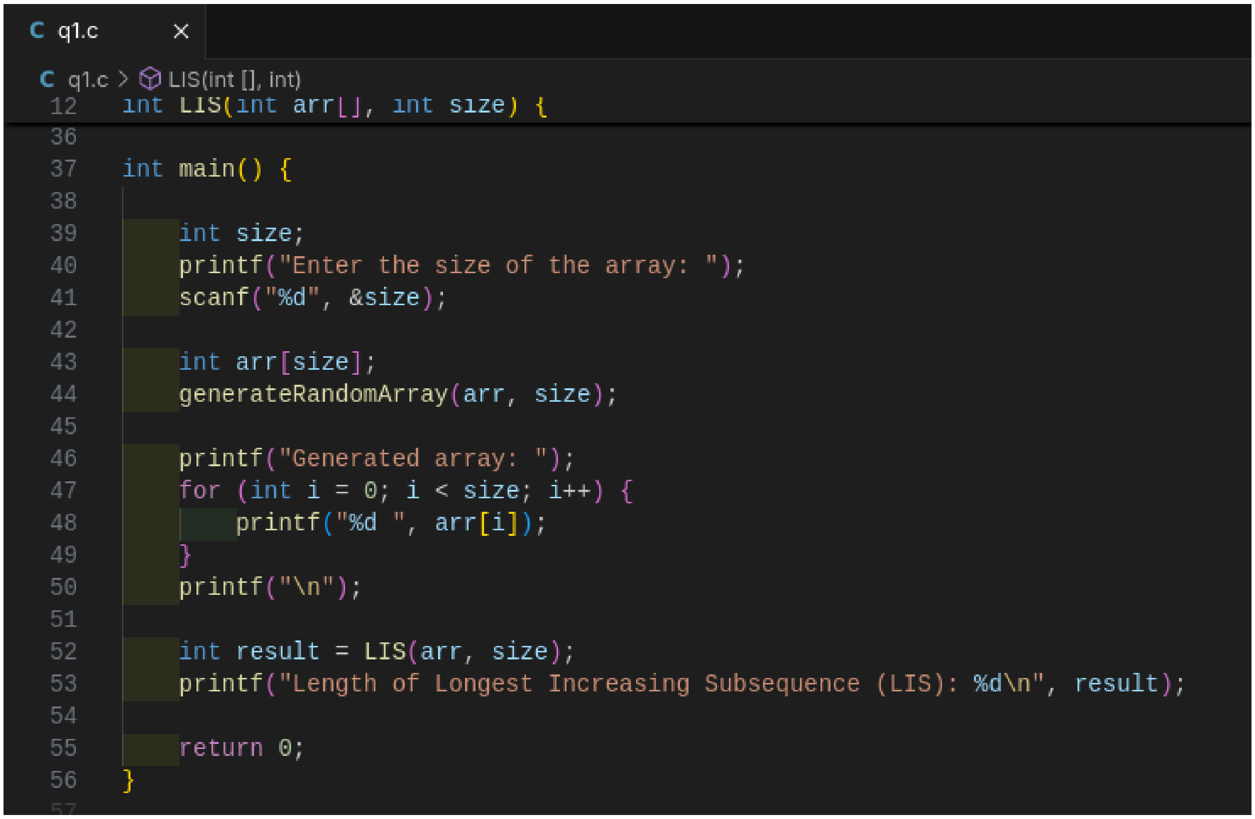
Arnav Jain - 220002018

Question 1

Code:

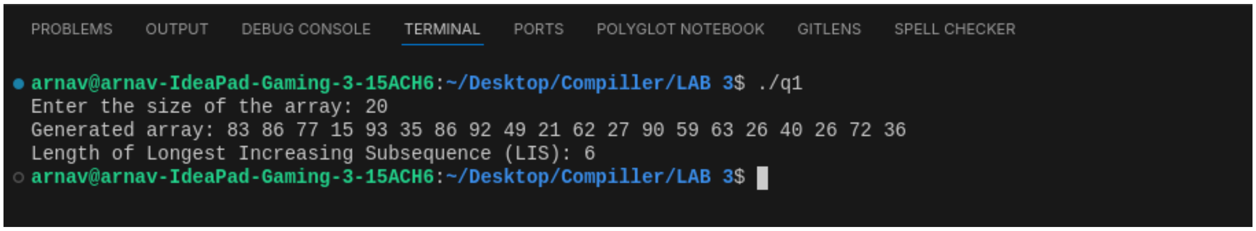


```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void generateRandomArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        arr[i] = rand() % 100;
    }
}


int LIS(int arr[], int size) {

    int lis[size];
    for (int i = 0; i < size; i++) {
        lis[i] = 1;
    }

    for (int i = 1; i < size; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[i] > arr[j] && lis[i] < lis[j] + 1) {
                lis[i] = lis[j] + 1;
            }
        }
    }

    int maxLIS = 0;
    for (int i = 0; i < size; i++) {
        if (lis[i] > maxLIS) {
            maxLIS = lis[i];
        }
    }

    return maxLIS;
}
```



```c
int LIS(int arr[], int size) {

int main() {

    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];
    generateRandomArray(arr, size);

    printf("Generated array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    int result = LIS(arr, size);
    printf("Length of Longest Increasing Subsequence (LIS): %d\n", result);

    return 0;
}
```

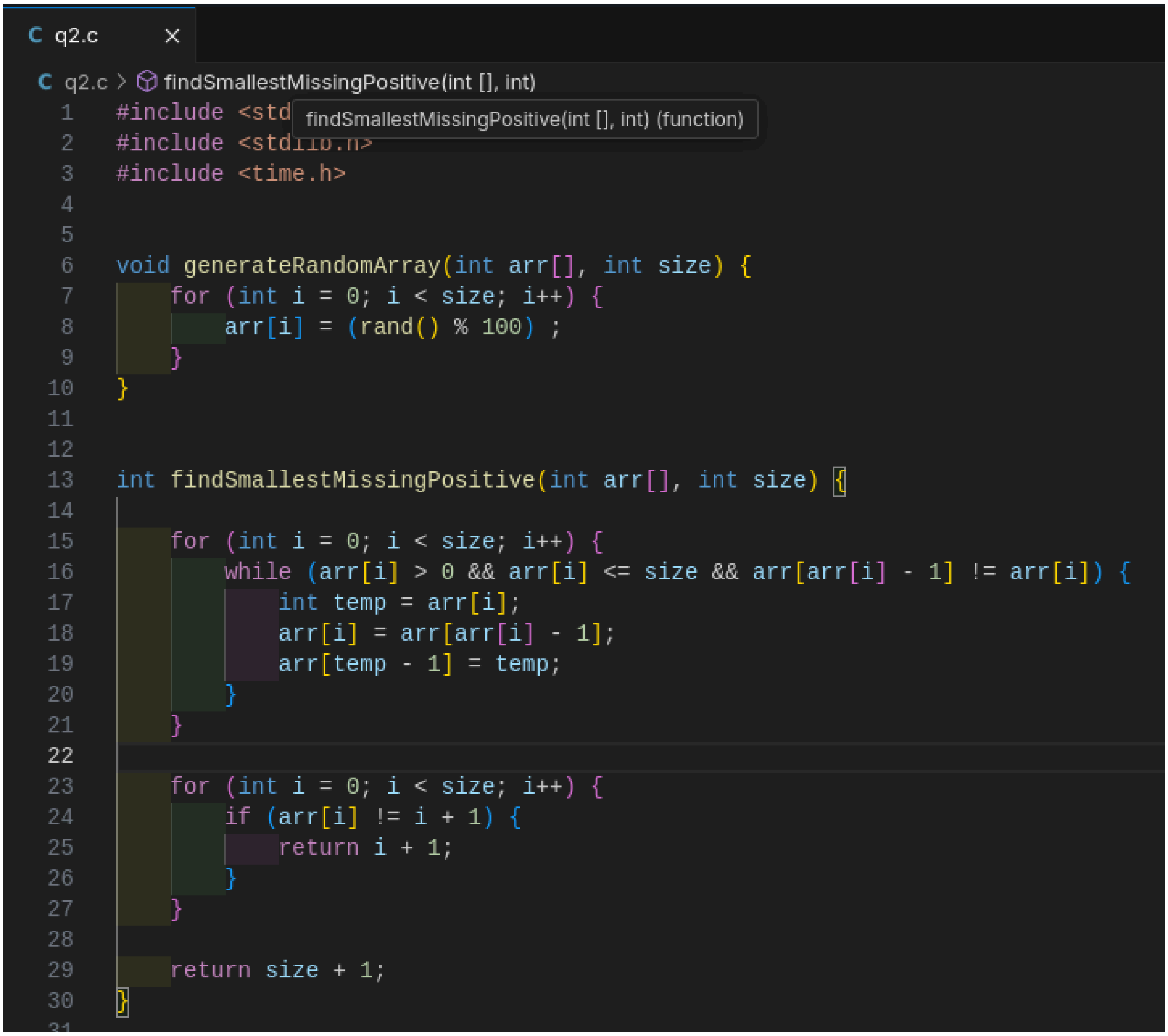
Result:



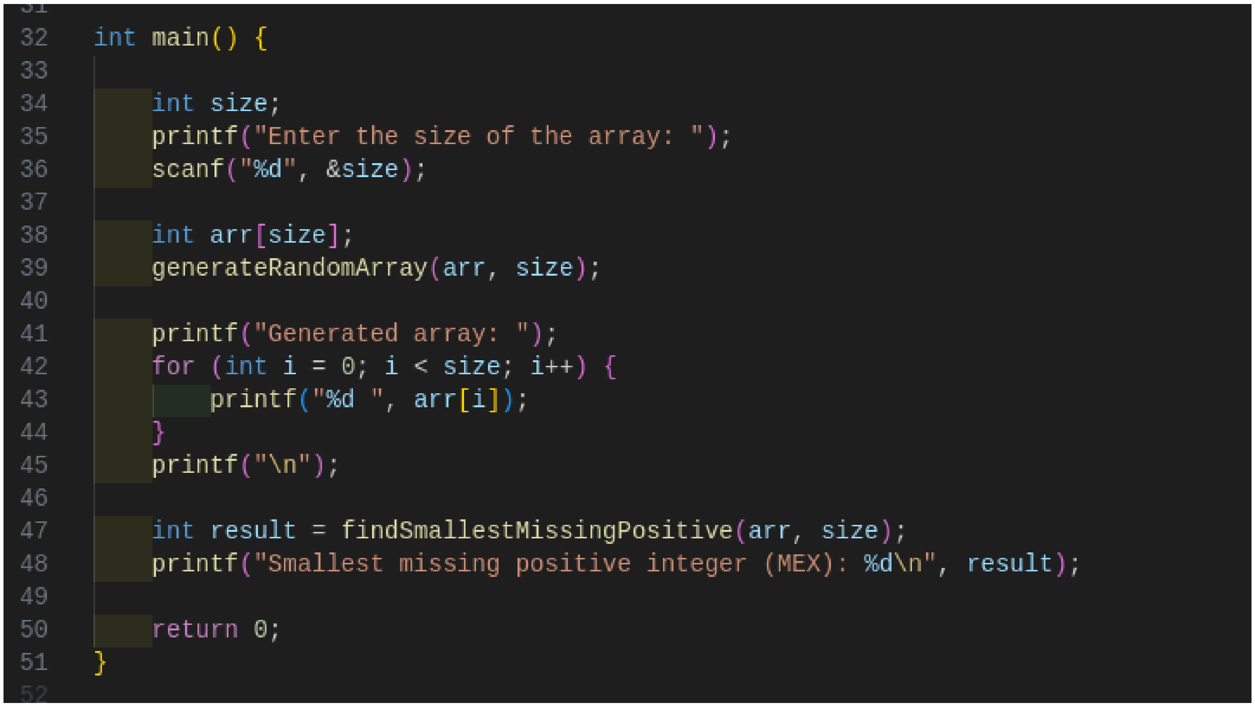
```
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 3$ ./q1
  Enter the size of the array: 20
  Generated array: 83 86 77 15 93 35 86 92 49 21 62 27 90 59 63 26 40 26 72 36
  Length of Longest Increasing Subsequence (LIS): 6
○ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 3$ █
```

Question 2

Code:

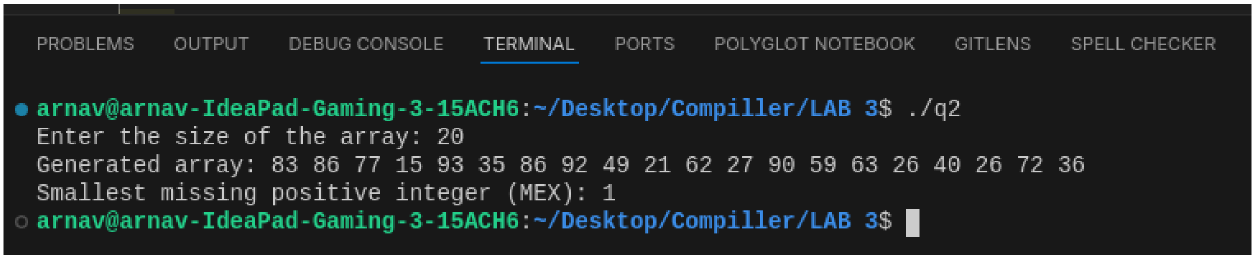

C q2.c ×
C q2.c > findSmallestMissingPositive(int [], int)
findSmallestMissingPositive(int [], int) (function)

```c
#include <std
#include <stdlib.h>
#include <time.h>


void generateRandomArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        arr[i] = (rand() % 100) ;
    }
}


int findSmallestMissingPositive(int arr[], int size) {

    for (int i = 0; i < size; i++) {
        while (arr[i] > 0 && arr[i] <= size && arr[arr[i] - 1] != arr[i]) {
            int temp = arr[i];
            arr[i] = arr[arr[i] - 1];
            arr[temp - 1] = temp;
        }
    }

    for (int i = 0; i < size; i++) {
        if (arr[i] != i + 1) {
            return i + 1;
        }
    }

    return size + 1;
}
```



```c
int main() {

    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];
    generateRandomArray(arr, size);

    printf("Generated array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    int result = findSmallestMissingPositive(arr, size);
    printf("Smallest missing positive integer (MEX): %d\n", result);

    return 0;
}
```

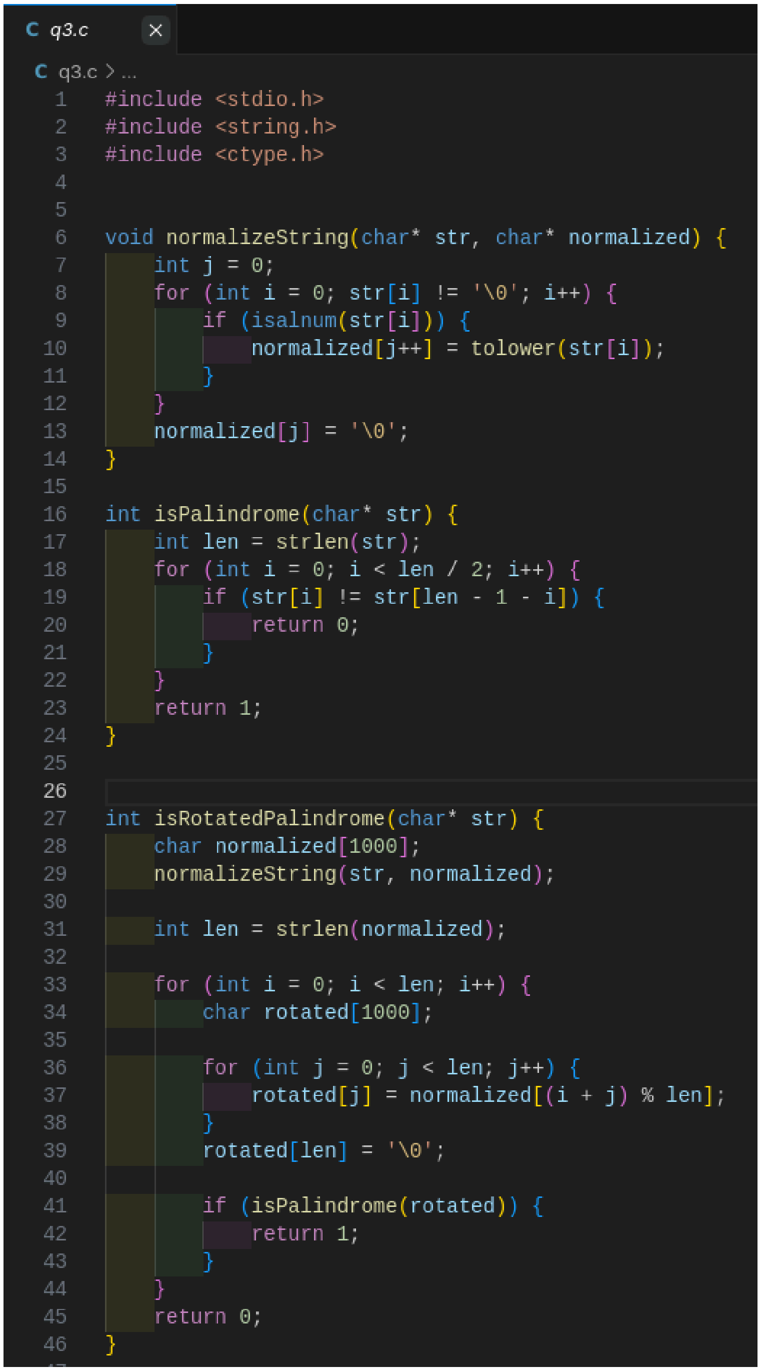
Result:



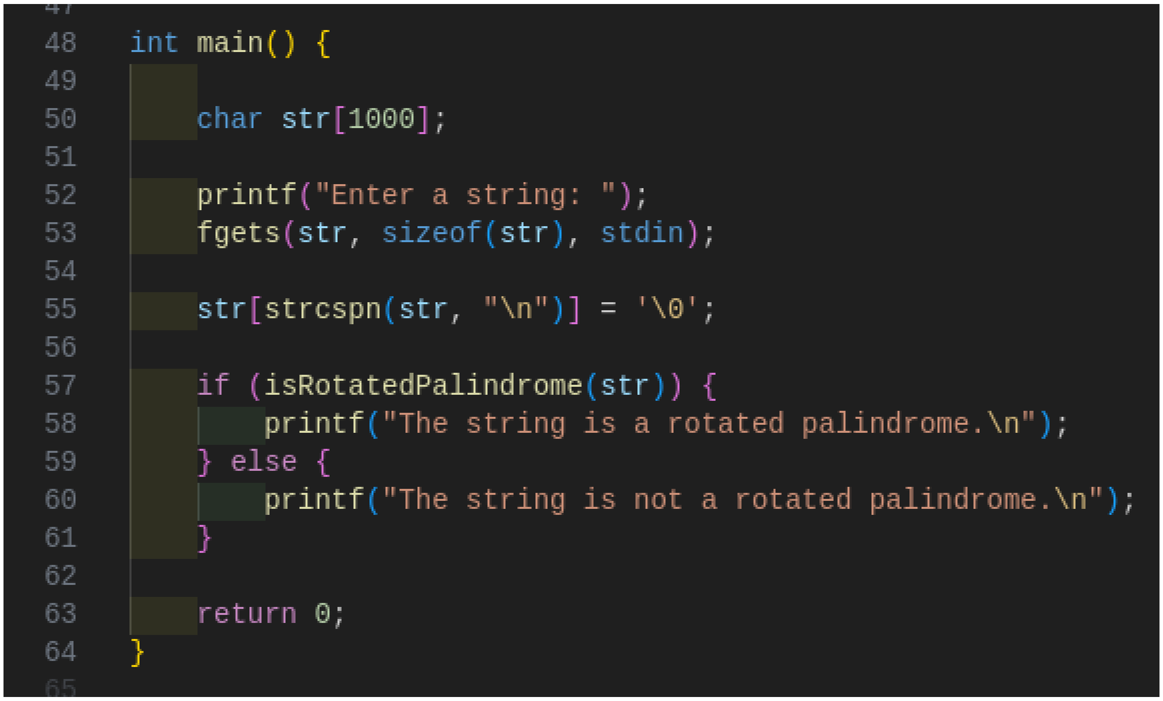
```
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 3$ ./q2
  Enter the size of the array: 20
  Generated array: 83 86 77 15 93 35 86 92 49 21 62 27 90 59 63 26 40 26 72 36
  Smallest missing positive integer (MEX): 1
○ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller/LAB 3$
```

Question 3

Code:

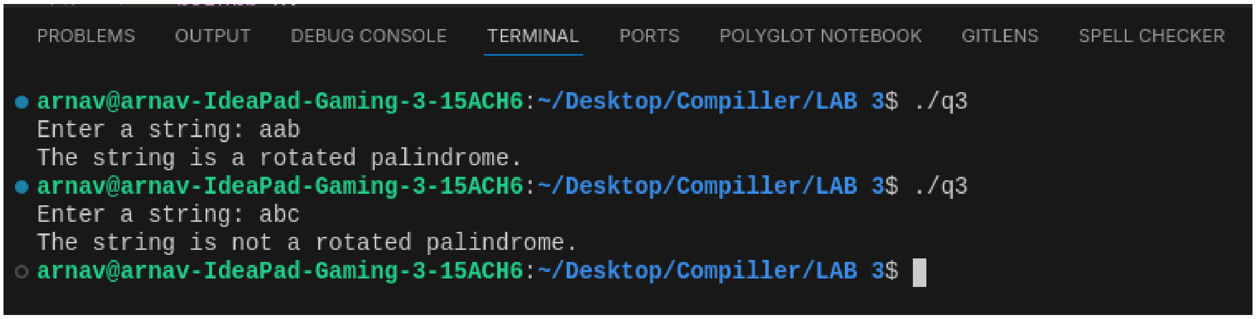


```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>


void normalizeString(char* str, char* normalized) {
    int j = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (isalnum(str[i])) {
            normalized[j++] = tolower(str[i]);
        }
    }
    normalized[j] = '\0';
}

int isPalindrome(char* str) {
    int len = strlen(str);
    for (int i = 0; i < len / 2; i++) {
        if (str[i] != str[len - 1 - i]) {
            return 0;
        }
    }
    return 1;
}


int isRotatedPalindrome(char* str) {
    char normalized[1000];
    normalizeString(str, normalized);

    int len = strlen(normalized);

    for (int i = 0; i < len; i++) {
        char rotated[1000];

        for (int j = 0; j < len; j++) {
            rotated[j] = normalized[(i + j) % len];
        }
        rotated[len] = '\0';

        if (isPalindrome(rotated)) {
            return 1;
        }
    }
    return 0;
}
```



```c
int main() {

    char str[1000];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    str[strcspn(str, "\n")] = '\0';

    if (isRotatedPalindrome(str)) {
        printf("The string is a rotated palindrome.\n");
    } else {
        printf("The string is not a rotated palindrome.\n");
    }

    return 0;
}
```

Result:



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

GitHub

https://github.com/arnavjain2710/Compiller-Techniques/tree/main/LAB%203