Introduction to Computer Science
Laboratory 10

1. Finish the exercises present in the 10^{th} tutorial.

2. Identify the issues in the following C programs. Then fix the programs, avoiding the use of global variables while maintaining the original intended functionality:

   A.

```c
#include <stdio.h>
int n;
int i;
int prime = 1;

void isPrime() {
        i = 2;
        while(i < n && prime) {
                if(n % 1 == 0) {
                        prime = 0;
                }
                i++;
        }
}

int main() {
        scanf("%d", &n);
        isPrime();
        if(prime) {
                printf("The number %d is prime\n", n);
        }
        else {
                printf("The number %d is not prime\n", n);
        }
        return 0;
}
```

   B.

```c
#include <stdio.h>
#define MAX_SIZE 100


int evens;
int odds;
int arr[MAX_SIZE];
int size;
int result;
int i;
int n;
```

```
void isEven() {
        if(n%2 == 0)
                result = 1;
        else
                result = 0;
}

void moreEvens() {
        for (i = 0; i<size; i++) {
                n = arr[i];
                isEven();
                if(result == 1) {
                        evens++;
                }
                else {
                        odds++;
                }
        }
        result = evens > odds;
}


int main() {
        scanf("%d", &size);
        for(i = 0; i<size; i++)
                scanf("%d", &arr[i]);
        i=0;
        evens=0;
        odds=0;
        moreEvens();
        if(result) {
                printf("There is more evens than odds\n");
        }
        else {
                printf("There is more or equal odds than evens\n");
        }
        return 0;
}
```

3. For the following C functions over arrays, add the const modifier where it is needed.

A.

```
void loadArray(int a[], int size) {
        for(int i = 0; i<size; i++) {
                a[i] = i;
        }
}
```

B.

```
void copy(int a[], int b[], int size) {
        for(int i = 0; i<size; i++) {
                b[i] = a[i];
        }
}
```

C.

```
void powArray(int a[], int size) {
        for(int i = 0; i<size; i++) {
                a[i] = a[i]*a[i];
        }
}
```

D.

```
void sumArrays(int a[], int b[], int c[], int size) {
        for(int i = 0; i<size; i++) {
                c[i] = a[i]+b[ci];
        }
}
```

4. Implement a C function that given 2 strings, determines if they are equal or not. Then write a main function that takes 2 strings from standard input and calls the previously defined function.

5. Design a function that given a string, calculates their length. Write a main function that takes as an input a string from command line and calculates the string's length.

6. Write a function that given a string s, calculates the reverse of s and stores it in the same string. **This function cannot use additional arrays**. After, write a main function that takes a string from standard input, prints it, and prints the reverse of the input.

7. Write a C function that calculates the n^{th} power of 2 using an array. **This array must have a length of (n+1) for every n.** Then write a main function that takes n as an input from command line and calculates the n^{th} power of 2.