EXERCISE 2

Task-1

```c
#include <stdio.h>
// TASK1
int main() {
    int n, i, sum = 0;

    printf("Enter n: ");
    scanf("%d", &n);

    // --- Using for loop ---
    printf("\nUsing for loop:\n");
    sum = 0;
    for (i = 1; i <= 2*n; i += 2) {  // Sirf odd numbers print karne ke liye i+=2
% ka bhi use kr skte
        printf("%d ", i);
        sum += i;
    }
    printf("\nSum = %d\n", sum);

    // --- Using while loop ---
    printf("\nUsing while loop:\n");
    sum = 0;
    i = 1;
    int count = 0;
    while (count < n) {
        printf("%d ", i);
        sum += i;
        i += 2;
        count++;
    }
    printf("\nSum = %d\n", sum);

    // --- Using do-while loop can be done wit any other loop as well
    printf("\nUsing do-while loop:\n");
    sum = 0;
    i = 1;
    count = 0;
    do {
        printf("%d ", i);
        sum += i;
```

```
        i += 2;
        count++;
    } while (count < n);
    printf("\nSum = %d\n", sum);

    return 0;
}
```

```
PS C:\Users\NB\Desktop\cewTSKS\lab1> cd "c:\Users\NB\Desktop\cewTSKS\lab2\" ; if ($?) { g++ task1.cpp
  -o task1 } ; if ($?) { .\task1 }
Enter n: 5

Using for loop:
1 3 5 7 9
Sum = 25

Using while loop:
1 3 5 7 9
Sum = 25

Using do-while loop:
1 3 5 7 9
Sum = 25
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

Task-2

```
#include <stdio.h>

// remember the intuition
int main() {
    int i, j, space, rows;
    printf("Enter number of rows for pyramid: ");
    scanf("%d", &rows);
    for (i = 1; i <= rows; i++) {
        for (space = 1; space <= rows - i; space++) {
            printf(" ");
        }
        for (j = 1; j <= (2 * i - 1); j++) {
            printf("*");
        }

        printf("\n");
    }
    return 0;
```

```
PS C:\Users\NB\Desktop\cewTSKS\lab2> cd "c:\Users\NB\Desktop\cewTSKS\lab2\" ; if ($?) { g++ tempCodeR
unnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number of rows for pyramid: 5
    *
   ***
  *****
 *******
*********
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

Task-3

```c
#include <stdio.h>

int main() {
    char str1[100], str2[100];
    int i = 0, flag = 0;

    printf("Enter first string: ");
    gets(str1);
    printf("Enter second string: ");
    gets(str2);
// scanf can also be used but gets better

    while (str1[i] != '\0' && str2[i] != '\0') {
        if (str1[i] != str2[i]) {
            flag = 1;  // Agar koi character different mil jaye
            break;
        }
        i++;
    }

    if (flag == 0 && str1[i] == '\0' && str2[i] == '\0') {
        printf("Strings are equal.\n");
    } else {
        printf("Strings are not equal.\n");
    }

    return 0;
}
```

```
Enter first string: horse
Enter second string: horse
Strings are equal.
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

```
Enter first string: hello
Enter second string: bye
Strings are not equal.
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

Task-4

```c
#include <stdio.h>

// Task4
int main() {
    char str[200];
    int i = 0;

    printf("Enter a sentence: ");
    gets(str);

    while (str[i] != '\0') {
        if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] = str[i] - 32;
        } else if (str[i] >= 'A' && str[i] <= 'Z') {
            str[i] = str[i] + 32; // Uppercase ko lowercase
        }
        i++;
    }

    printf("Converted sentence: %s\n", str);

    return 0;
}
```

```
PS C:\Users\NB\Desktop\cewTSKS\lab2> cd "c:\Users\NB\Desktop\cewTSKS\lab2\" ; if ($?) { g++ task4.cpp
 -o task4 } ; if ($?) { .\task4 }
task4.cpp: In function 'int main()':
task4.cpp:9:9: warning: call to 'gets' declared with attribute warning: Using gets() is always unsafe
 - use fgets() instead [-Wattribute-warning]
    9 |     gets(str);
      |     ~~~~^~~~~
Enter a sentence: hello
Converted sentence: HELLO
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

Task-5

```c
#include <stdio.h>
int main() {
    int n, i, j, count;
    int arr[100];

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

    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Unique elements: ");
    for (i = 0; i < n; i++) {
        count = 0;
        for (j = 0; j < n; j++) {
            if (arr[i] == arr[j] && i != j) {
                count++;
                break;
            }
        }
        if (count == 0) {
            printf("%d ", arr[i]);
        }
    }
    printf("\n");

    return 0;
}
```

```
PS C:\Users\NB\Desktop\cewTSKS\lab2> cd "c:\Users\NB\Desktop\cewTSKS\lab2\" ; if ($?) { g++ tempCodeR
unnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter size of array: 5
Enter 5 elements: 1 3 4 4 1
Unique elements: 3
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```

Task-6

```c
#include <stdio.h>
// rem struct node linked list
struct Distance {
    int feet;
    int inch;
};
int main() {
    struct Distance d1, d2, sum;
    printf("Enter 1st distance (feet inch): ");
    scanf("%d %d", &d1.feet, &d1.inch);
    printf("Enter 2nd distance (feet inch): ");
    scanf("%d %d", &d2.feet, &d2.inch);
    sum.feet = d1.feet + d2.feet;
    sum.inch = d1.inch + d2.inch;
// if >  12 inch make itr feet
    if (sum.inch >= 12) {
        sum.feet += sum.inch / 12;
        sum.inch = sum.inch % 12;
    }
    printf("Sum of distances = %d feet %d inch\n", sum.feet, sum.inch);
    return 0;}
```

```
PS C:\Users\NB\Desktop\cewTSKS\lab2> cd "c:\Users\NB\Desktop\cewTSKS\lab2\" ; if ($?) { g++ tempCodeR
unnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter 1st distance (feet inch): 5 6
Enter 2nd distance (feet inch): 6 3
Sum of distances = 11 feet 9 inch
PS C:\Users\NB\Desktop\cewTSKS\lab2>
```