

Review Loops (While and For)

CSE 1310 –Bhanu Jain
University of Texas at Arlington

Lab 2 Q6

- **Q6** Complete that program, by defining a `countOccurrences` function, and by modifying the existing `mostFrequentCharacter` function, to satisfy the following specs:
- Function **`countOccurrences`** takes two arguments, a string, and a character. It returns the number of times the character occurs in the string.
- Function **`mostFrequentCharacter`** takes a string as an argument. It returns the character that occurs the most times in the string. If multiple characters tie for occurring the most times, the function can return any one of those characters tying for most occurrences. Any of those return values will be considered correct.
- You are not allowed to change the main program in any way.

```
int main() {
    char text[21];
    while (true) {
        printf("Enter some text, or q to quit: ");
        gets(text);
        if (strcmp(text,"q")==0) {    break;    }
        if (strlen(text) == 0) {      break;      }
        char c = mostFrequentCharacter(text);
        int number = countOccurrences(text, c);
        printf("Most frequent character: '%c'\n", c);
        printf("Number of occurrences of '%c': %d\n\n", c, number);
    }
    printf("Exiting...\n");
    return 0;
}
```

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

..... countOccurrences(.....) {
    .....
}

..... mostFrequentCharacter(.....) {

}
```

Lab 2 Q6

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

int countOccurrences(char text[], char charToCount) {
    int counter = 0;
    for(int i=0;i<strlen(text);i++) {
        if(text[i] == charToCount) { counter++; }
    }
    return counter;
}

char mostFrequentCharacter(char text[]) {
    int max_counter = 0, counter=0;
    char max_char = 'a'; // the initial value of max_char makes no difference.

    for (int i = 0; i < strlen(text); i++) {
        char current = text[i];
        counter = countOccurrences(text, current);
        if (counter > max_counter) {
            max_counter = counter;
            max_char = current;
        }
    }
    return max_char;
}
```

```
int main() {
    char text[21];
    while (true) {
        printf("Enter some text, or q to quit: ");
        gets(text);
        if (strcmp(text,"q")==0) { break; }
        if (strlen(text) == 0) { break; }
        char c = mostFrequentCharacter(text);
        int number = countOccurrences(text, c);
        printf("Most frequent character: '%c'\n", c);
        printf("Number of occurrences of '%c': %d\n\n", c, number);
    }
    printf("Exiting...\n");
    return 0;
}
```

Designing a **while** loop

- When you design a **while** loop, you need to make sure that the loop will terminate exactly when needed, not before, and not after.
- You will need to define a boolean condition, that determines exactly when to stay in the loop and when to exit.
- You need to update variables within the body of the loop, as needed, to make sure that the boolean condition will evaluate to the right thing.

Q1

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers from 1 to N.
 - Produces an output as shown below:

```
Please enter an integer: 10
1
2
3
4
5
6
7
8
9
10
done with the while loop.
```

Q1

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers from 1 to N.
 - Produces an output as shown below:

```
#include <stdio.h>

int main() {
    int N;

    printf("Please enter an integer: ");
    scanf("%d", &N);
    int i = 1;
    while (i <= N)
    {
        printf("%d\n", i);
        i = i+1;
    }
    printf("done with the while loop.\n");
    return 0;
}
```

```
Please enter an integer: 10
1
2
3
4
5
6
7
8
9
10
done with the while loop.
```

Q2

- Write a program that:
 - Asks the user to enter an integer N .
 - Prints whether N is prime.
- Strategy for determining if N is prime:
 - For every number K between 2 and $N-1$, check if K divides N .

Q2

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints whether N is prime.
- Strategy for determining if N is prime:
 - For every number K between 2 and N-1, check if K divides N.

```
#include <stdio.h>
#include <stdbool.h>
int main() {
    int N;
    printf("Please enter an integer: ");
    scanf("%d", &N);
    bool N_is_prime = true;
    int i = 2;
    while (i < N)
    { if (N % i == 0)      { N_is_prime = false; break; }
      i++;
    }
    if (N_is_prime) {printf("%d is prime.\n", N); }
    else            {printf("%d is not prime.\n", N); }
    return 0;
}
```


Example of a **for** loop:

Q3 Printing Numbers from 1 to N

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers from 1 to N.

```
Please enter an integer: 10
1
2
3
4
5
6
7
8
9
10
done with the for loop.
```

Printing Numbers from 1 to N

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers from 1 to N.

```
#include <stdio.h>
int main() {
    int N;
    printf("Please enter an integer: ");
    scanf("%d", &N);
    for (int i = 1; i <= N; i++)
    {
        printf("%d\n", i);
    }
    printf("done with the for loop.\n");
    return 0;
}
```

```
#include <stdio.h>
int main() {
    int N;
    printf("Please enter an integer: ");
    scanf("%d", &N);
    int i = 1;
    while (i <= N)
    {
        printf("%d\n", i);    i = i+1;
    }
    printf("done with the while loop.\n");
    return 0;
}
```

Q4

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers that are less than N and multiples of 13.

```
Please enter an integer: 130
0
13
26
39
52
65
78
91
104
117
130
Printed all numbers between 0 and 130
that are multiples of 13.
```

Q4

- Write a program that:
 - Asks the user to enter an integer N.
 - Prints all integers that are less than N and multiples of 13.

```
Please enter an integer: 130
0
13
26
39
52
65
78
91
104
117
130
printed all numbers between 0 and 130
that are multiples of 13.
```

```
#include <stdio.h>
int main() {
    int N;
    printf("Please enter an integer: ");
    scanf("%d", &N);
    for (int i = 0; i <= N; i+=13)
    {
        printf("%d\n", i);
    }
    printf("printed all numbers between 0 and %d\n", N);
    printf("that are multiples of 13.\n", N);
    return 0;
}
```

Q4

- Write a program that:
 - Asks the user to enter integers N1 and N2.
 - Prints all integers that are less between N1 and N2 and are multiples of 13.

```
Please enter an integer N1: 22
Please enter an integer N2: 130
26
39
52
65
78
91
104
117
130
```

Q4

- Write a program that:
 - Asks the user to enter integers N1 and N2.
 - Prints all integers that are less between N1 and N2 and are multiples of 13.

```
Please enter an integer N1: 22
Please enter an integer N2: 130
26
39
52
65
78
91
104
117
130
```

```
#include <stdio.h>
int main() {
    int N1,N2;
    printf("Please enter an integer N1: ");
    scanf("%d",&N1);
    printf("Please enter an integer N2: ");
    scanf("%d",&N2);
    for (int i = N1; i <= N2; i++)
    {
        if (i%13==0) {printf("%d\n", i);}
    }
    printf("printed all numbers between %d and %d\n",N1,N2);
    printf("that are multiples of 13.\n" );
    return 0;
}
```

Q5

- Write a program that:
 - Asks the user to enter an integers N1.
 - Prints all integers counting downwards from N1 to 0.

```
Please enter an integer N1: 10
10
9
8
7
6
5
4
3
2
1
0
printed all numbers between 10 and 0.
```

Q5

- Write a program that:
 - Asks the user to enter an integers N1.
 - Prints all integers counting downwards

```
Please enter an integer N1: 10
10
9
8
7
6
5
4
3
2
1
0
printed all numbers between 10 and 0.
```

```
#include <stdio.h>
int main() {
    int N1;
    printf("Please enter an integer N1: ");
    scanf("%d", &N1);

    for (int i = N1; i >= 0; i--)
    {
        printf("%d\n", i);
    }
    printf("printed all numbers between %d and %d.\n", N1, 0);
    return 0;
}
```


Q6

- Write a program that:
 - Asks the user to enter a word.
 - Prints each letter of that word on a separate line.

```
Please enter a word: Orange
O
r
a
n
g
e
```

Q6

- Write a program that:
 - Asks the user to enter a word.
 - Prints each letter of that word on a separate line.

```
Please enter a word: Orange
O
r
a
n
g
e
```

```
#include <stdio.h>
#include <string.h>
int main() {
    char word[50];
    printf("Please enter a word: ");
    scanf("%s", word);
    for (int i = 0; i < strlen(word); i++)
    {
        printf("%c\n", word[i]);
    }
    return 0;
}
```

- Write a program that: **Q7**
 - Asks the user to enter a word.
 - Starting from the first letter, it prints every other letter of the word. The letters should be printed on the same line, **not** one per line.

```
Please enter a word: Orange
Oag
```

• Write a program that: Q7

- Asks the user to enter a word.
- Starting from the first letter, it prints every other letter of the word. The letters should be printed on the same line, **not** one per line.

```
Please enter a word: Orange
Oag
```

```
#include <stdio.h>
int main() {
    char word[50];
    printf("Please enter a word: ");
    scanf("%s", word);
    for (int i = 0; i < strlen(word); i+=2)
    {
        printf("%c", word[i]);
    }
    return 0;
}
```

- Write a program that: **Q8**
 - Asks the user to enter a word.
 - Prints the letters of the string backwards. The letters should be printed on the same line, **not** one per line.

```
Please enter a word: Orange  
egnarO
```

• Write a program that: Q8

– Asks the user to enter a word.

```
Please enter a word: Orange  
egnarO
```

– Prints the letters of the string backwards. The letters should be printed on the same line, **not** one per line.

```
#include <stdio.h>
int main() {
    char word[50];
    printf("Please enter a word: ");
    scanf("%s", word);
    int i = 0;
    for ( i = strlen(word) - 1; i >= 0; i--)
    {
        printf("%c", word[i]);
    }
    printf("\n");
    return 0;
}
```

```
#include <stdio.h>
int main() {
    char word[50];
    printf("Please enter a word: ");
    scanf("%s", word);
    int i = strlen(word) - 1;
    while (i >= 0) {
        printf("%c", word[i]);
        i--;
    }
    printf("\n");
    return 0;
}
```

Q9

- Write a program that:
 - Asks the user to enter some text.
 - Counts the number of times the letter 'a' appears in the text.

```
Please enter a word: Narnia
The letter a occurs 2 times.
```

Q9

- Write a program that:
 - Asks the user to enter some text.
 - Counts the number of times the letter 'a' appears in the text.

```
Please enter a word: Narnia
The letter a occurs 2 times.
```

```
#include <stdio.h>
#include <string.h>
int main() {
    char word[50];
    printf("Please enter a word: ");
    scanf("%s", word);
    int counter = 0;
    for (int i = 0; i < strlen(word); i++)
    {
        char c = word[i];
        if (c == 'a') { counter++; }
    }
    printf("The letter a occurs %d times.\n", counter);
    return 0;
}
```


Q10

- Write a program that:
 - Asks the user to enter an integers N1.
 - Prints the first number that is greater than N1 and divisible by 13.

```
Please enter a number: 15  
26 is the first integer >= 15 that is divisible by 13.
```

Q10

- Write a program that:
 - Asks the user to enter an integer N1
 - Prints the first number that is greater than N1 and divisible by 13.

```
Please enter a number: 15
26 is the first integer >= 15 that is divisible by 13.
```

```
#include <stdio.h>
#include <stdbool.h>
int main() {
    int N;
    printf("Please enter a number: ");
    scanf("%d", &N);
    int i = N;
    while(true) {
        if (i % 13 == 0) {
            printf("%d is the first integer >= %d that is divisible by 13.\n", i, N);
            break; //leave the while loop
        }
        i++;
    }
    return 0;
}
```

Q10

- Write a program that:
 - Asks the user to enter an integer N1
 - Prints the first number that is greater than N1 and divisible by 13.

```
Please enter a number: 15
26 is the first integer >= 15 that is divisible by 13.
```

```
#include <stdio.h>
#include <stdbool.h>
int main() {
    int N;
    printf("Please enter a number: ");
    scanf("%d", &N);
    int i = N;
    for (;;) { //infinite for loop
        if (i % 13 == 0) {
            printf("%d is the first integer >= %d that is divisible by 13.\n", i, N);
            break;
        }
        i++;
    }
    return 0;
}
```

Nested Loops

- A loop can be part of another loop. Such a loop is called a **nested loop**.
- Example 1: Print out the 10x10 multiplication table.

Q11

- Write a program that:
 - Prints the number 1 through 10 as shown here.

```
1 2 3 4 5 6 7 8 9 10
```

Q11

- Write a program that:
 - Prints the number 1 through 10 as shown here.

1 2 3 4 5 6 7 8 9 10

```
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
    {
        printf("%d ", i);
    }
    return 0;
}
```

Q12

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 10 as shown here.

```
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
```

Q12

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 10 as shown here.

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

```
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
```


Q13

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 100 as shown here.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Q13

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 100 as shown here.

```
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%3d ", i*j);
        }
        printf("\n");
    }
    return 0;
}
```

Q14

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 100 as shown here.

```
1
2  4
3  6  9
4  8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
```

Q14

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 100 as shown

```
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {if (j > i) { break; }
        printf("%3d ", i*j);
        }
        printf("\n");
    }
    return 0;
}
```

```
1
2  4
3  6  9
4  8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
```

Q14

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 100 as shown here.

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {if (j > i) { break; }
        printf("%3d ", i*j);
        }
        printf("\n");
    }
    return 0;
}
```

1									
2	4								
3	6	9							
4	8	12	16						
5	10	15	20	25					
6	12	18	24	30	36				
7	14	21	28	35	42	49			
8	16	24	32	40	48	56	64		
9	18	27	36	45	54	63	72	81	
10	20	30	40	50	60	70	80	90	100

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%3d ", i*j);
        }
        printf("\n");
    }
    return 0;
}
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Q15

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 50 as shown here.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50

Q15

- Write a program that:
 - Uses nested for loops
 - Prints the number 1 through 50 as shown here.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50

```
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        { printf("%3d ", i*j); }
        printf("\n");
        if (i == 5) {break;}
    }
    return 0;
}
```

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%3d ", i*j);
        }
        printf("\n");

        if (i == 5) {break;}
    }

    return 0;
}
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}
```

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            if (j > i) { break; }

            printf("%3d ", i*j);
        }

        printf("\n");
    }

    return 0;
}
```

1									
2	4								
3	6	9							
4	8	12	16						
5	10	15	20	25					
6	12	18	24	30	36				
7	14	21	28	35	42	49			
8	16	24	32	40	48	56	64		
9	18	27	36	45	54	63	72	81	
10	20	30	40	50	60	70	80	90	100

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++)
    {
        for (int j = 1; j <= 10; j++)
        {
            printf("%3d ", i*j);
        }

        printf("\n");
    }

    return 0;
}
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Q16

- Write a program that:
 - Accepts the radius of a circle.
 - Computes and prints the circumference and the area of the circle.

```
Please enter the radius: 5
The circumference is 31.42.
The area is 78.54.
```

Q16

- Write a program that:
 - Accepts the radius of a circle/
 - Computes and prints the circumference and the area of the circle.

```
Please enter the radius: 5
The circumference is 31.42.
The area is 78.54.
```

```
#include <stdio.h>
#include <math.h>
int main() {
    double radius;
    printf("Please enter the radius: ");
    scanf("%lf",&radius);
    double circumference = 2 * M_PI * radius;
    double area = M_PI * pow(radius, 2);
    printf("The circumference is %.2f.\n", circumference);
    printf("The area is %.2f.\n", area);
    return 0;
}
```

Q16

- Write a program that:
 - Has an **infinite** loop.
 - Accepts the radius of a circle.
 - Computes and prints the circumference and the area of the circle.

```
Please enter the radius: 5
The circumference is 31.42.
The area is 78.54.

Please enter the radius: 6
The circumference is 37.70.
The area is 113.10.

Please enter the radius: 7
The circumference is 43.98.
The area is 153.94.

Please enter the radius:
```

Q16

- Write a program that:
 - Has an infinite loop.
 - Accepts the radius of a circle.
 - Computes and prints the circumference and the area of the circle.

```
Please enter the radius: 5
The circumference is 31.42.
The area is 78.54.

Please enter the radius: 6
The circumference is 37.70.
The area is 113.10.

Please enter the radius: 7
The circumference is 43.98.
The area is 153.94.

Please enter the radius:
```

```
#include <stdio.h>
#include <math.h>
#include <stdbool.h>
int main() {
    double radius;
    while (true){
        printf("Please enter the radius: ");
        scanf("%lf",&radius);
        double circumference = 2 * M_PI * radius;
        double area = M_PI * pow(radius, 2);
        printf("The circumference is %.2f.\n", circumference);
        printf("The area is %.2f.\n", area);
        printf("\n\n");
    }
    return 0;
```

Q17

- Write a program that:
 - Has an **infinite** loop.
 - **Accepts the radius** of a circle **or -1 to quit** the program.
 - Computes and prints the circumference and the area of the circle.

```
Enter the circle radius, or -1 to quit: 5
The circumference is 31.42.
The area is 78.54.

Enter the circle radius, or -1 to quit: 6
The circumference is 37.70.
The area is 113.10.

Enter the circle radius, or -1 to quit: -1
Exiting...
```

Q17

```
Enter the circle radius, or -1 to quit: 5
The circumference is 31.42.
The area is 78.54.

Enter the circle radius, or -1 to quit: 6
The circumference is 37.70.
The area is 113.10.

Enter the circle radius, or -1 to quit: -1
Exiting...
```

- Write a program that:
 - Has an **infinite** loop.
 - Accepts the radius** of a circle **or -1 to quit** the program.
 - Computes and prints the circumference and the area of the circle.

```
#include <stdio.h>
#include <math.h>
#include <stdbool.h>
int main() {
    double radius;
    while (true){
        printf("Enter the circle radius, or -1 to quit: ");
        scanf("%lf",&radius);
        if (radius == -1) {
            printf("\nExiting...\n"); break;
        }
        double circumference = 2 * M_PI * radius;
        double area = M_PI * pow(radius, 2);
        printf("The circumference is %.2f.\n", circumference);
        printf("The area is %.2f.\n", area);
    }
    return 0;
}
```

Example Programs

- Summing integers from 1 to N, and variations.
 - Summing squares.
 - Summing multiples of 7.
 - Summing primes.
- Printing divisors of a number.
- Removing spaces, dashes, parentheses from a phone number (or a credit card number).
- Printing a pyramid using the * character.