Q.1. Write a C program to print the following pattern for N number of rows given by user

12345
1234
123
12
1

main.c

Save Run

```c
#include <stdio.h>
// 12345
// 1234
// 123
// 12
// 1
int main()
{
    int n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);

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

    return 0;
}
```

Output

```
/tmp/BByKiXthMl.o
Enter the number of rows: 7
1234567
123456
12345
1234
123
12
1
```

Q.1. Write a C program to print the following pattern for N number of rows given by user

```
   *
  **
 ***
****
*****
```

main.c

Save Run

```c
#include <stdio.h>
//     *
//    **
//   ***
//  ****
// *****
int main()
{
    int n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);

    for (int i = 0; i <= n; i++)
    {
        for (int j = 1; j <= (n - i); j++)
        {
            printf(" ");
        }
        for (int k = 0; k <= i; k++)
        {
            printf("*");
        }

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

Output

```
/tmp/ZKwOqnYQjN.o
Enter the number of rows: 7
       *
      **
     ***
    ****
   *****
  ******
 *******
********
```

Q.1. Write a C program to print the following pattern:

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

main.c

Save Run

```c
#include <stdio.h>
//     1
//   2 3 4
// 5 6 7 8 9
int main()
{
    int count = 1;

    for (int i = 1; i <= 3; i++)
    {
        for (int j = 0; j <= (2 - i); j++)
        {
            printf(" ");
        }
        for (int k = 1; k <= (2 * i - 1); k++)
        {
            printf("%d", count++);
        }

        printf("\n");
    }

    return 0;
}
```

Output

```
/tmp/5TrO4CsT3l.o
  1
 234
56789
```

Q.2. Write a C program to check whether a number is palindrome or not.

main.c

Save Run

```c
#include <stdio.h>
//A palindrome is a number that is the same when reversed
int isPalindrome(int num) {
    int reversedNum = 0, originalNum = num;

    while (num != 0) {
        int remainder = num % 10;
        reversedNum = reversedNum * 10 + remainder;
        num /= 10;
    }

    if (originalNum == reversedNum) {
        return 1; // Palindrome
    } else {
        return 0; // Not a palindrome
    }
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (isPalindrome(num)) {
        printf("%d is a palindrome.\n", num);
    } else {
        printf("%d is not a palindrome.\n", num);
    }

    return 0;
}
```

Output

```
/tmp/xtOU9zWMe0.o
Enter a number: 1536654
1536654 is not a palindrome.

/tmp/F4z6E9xZjX.o
Enter a number: 12321
12321 is a palindrome.
```

Q.3. Write a C program to find sum of first and last digit of a number.

main.c

Save Run

```c
#include <stdio.h>

int main()
{
    int number, firstDigit, lastDigit, sum;

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

    // Extracting the first digit
    firstDigit = number;
    while (firstDigit >= 10)
    {
        firstDigit /= 10;
    }

    // Extracting the last digit
    lastDigit = number % 10;

    // Calculate the sum
    sum = firstDigit + lastDigit;

    printf("Sum of first and last digit: %d\n", sum);

    return 0;
}
```

Output

```
/tmp/dFJ5mFBS6H.o
Enter a number: 234234
Sum of first and last digit: 6

/tmp/lvRpnkvrDZ.o
Enter a number: 0
Sum of first and last digit: 0

/tmp/5dDOaokElB.o
Enter a number: 7
Sum of first and last digit: 14

/tmp/im2aQyMIeJ.o
Enter a number: 100
Sum of first and last digit: 1
```

Q.4. Write a C program to check whether a number is Armstrong number or not.

main.c

Save Run

```c
#include <stdio.h>
#include <math.h>
//An Armstrong number is a number that is the sum of its own digits each raised to the power of the number of digits
int isArmstrong(int num)
{
    int originalNum, remainder, result = 0, n = 0;
    originalNum = num;
    // count the number of digits
    while (originalNum != 0)
    {
        originalNum /= 10;
        ++n;
    }
    originalNum = num;
    // calculate the result
    while (originalNum != 0)
    {
        remainder = originalNum % 10;
        result += pow(remainder, n);
        originalNum /= 10;
    }
    // check if the number is Armstrong
    if (result == num)
        return 1;
    else
        return 0;
}

int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (isArmstrong(num))
        printf("%d is an Armstrong number.", num);
    else
        printf("%d is not an Armstrong number.", num);
    return 0;
}
```

Output

```
/tmp/RTaLTaqh2M.o
Enter a number: 153
153 is an Armstrong number.

/tmp/rnu8V9sI4k.o
Enter a number: 100
100 is not an Armstrong number.

/tmp/h0bK35dXkL.o
Enter a number: 0
0 is an Armstrong number.

/tmp/KQEO068pOr.o
Enter a number: 4343
4343 is not an Armstrong number.

/tmp/uqCnHqJ50b.o
Enter a number: 1634
1634 is an Armstrong number.
```

Q.5. Write a C program to calculate product of digits of a number.

main.c

Save Run

```c
#include <stdio.h>

int main() {
    int number, product = 1;

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

    while (number != 0) {
        int digit = number % 10;
        product *= digit;
        number /= 10;
    }

    printf("Product of digits: %d\n", product);

    return 0;
}
```

Output

```
/tmp/jtiIw6ZOba.o
Enter a number: 432
Product of digits: 24

/tmp/dhvo3TW50Z.o
Enter a number: 9
Product of digits: 9

/tmp/3nKXaUMPom.o
Enter a number: 3230
Product of digits: 0
```

Q.6. Write a C Program to print following pattern:

I :

main.c

Save Run

```c
#include <stdio.h>
int main()
{
    int n, j, i;
    printf("Enter number of rows :");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n - i; j++)
            printf(" ");

        for (j = 1; j <= 2 * i - 1; j++)
            printf("*");

        printf("\n");
    }
    for (i = 1; i <= n - 1; i++)
    {
        for (j = 1; j <= i; j++)
            printf(" ");

        for (j = 1; j <= 2 * (n - i) - 1; j++)
            printf("*");

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

Output

```
/tmp/kUqFOknHty.o
Enter number of rows :7
      *
     ***
    *****
   *******
  *********
 ***********
*************
 ***********
  *********
   *******
    *****
     ***
      *
```

II :

main.c

Save Run

```c
#include <stdio.h>
int main()
{
    int i, j, n;
    printf("Enter value of n : ");
    scanf("%d", &n);

    for(i=1; i<n; i++)
    {
        for(j=1; j<=(n-i); j++)
        {
            printf(" ");
        }
        for(j=i; j<=n; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    for(i=1; i<=n; i++)
    {
        for(j=1; j<i; j++)
        {
            printf(" ");
        }
        for(j=1; j<=i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
/tmp/hC3G8IJiwa.o
Enter value of n : 10
         **********
        *********
       ********
      *******
     ******
    *****
   ****
  ***
 **
*
 **
  ***
   ****
    *****
     ******
      *******
       ********
        *********
         **********
```

III :

main.c

Save Run

```c
#include <stdio.h>

int main() {
    int numLines;
    printf("Enter the number of lines: ");
    scanf("%d", &numLines);

    for (int i = 1; i <= numLines; i++) {
        for (int j = 1; j <= i; j++) {
            if (j % 2 == 0) {
                printf("0");
            } else {
                printf("1");
            }
        }
        printf("\n");
    }

    return 0;
}

```

Output

```
/tmp/YBgcSdBCKC.o
Enter the number of lines: 7
1
10
101
1010
10101
101010
1010101
```