Table of Contents

1. Basic programming exercises to practice for beginners.

1. Write a C program to enter two numbers and find their sum.
2. Write a C program to enter two numbers and perform all arithmetic operations.
3. Write a C program to enter length and breadth of a rectangle and find its perimeter.
4. Write a C program to enter length and breadth of a rectangle and find its area.
5. Write a C program to enter radius of a circle and find its diameter, circumference and area.
6. Write a C program to enter length in centimeter and convert it into meter and kilometer.
7. Write a C program to enter temperature in °Celsius and convert it into °Fahrenheit.
8. Write a C program to enter temperature in Fahrenheit(°F) and convert it into Celsius(°C)
9. Write a C program to convert days into years, weeks and days.
10. Write a C program to find power of any number x ^ y.
11. Write a C program to enter any number and calculate its square root.
12. Write a C program to enter two angles of a triangle and find the third angle.
13. Write a C program to enter base and height of a triangle and find its area.
14. Write a C program to calculate area of an equilateral triangle.
15. Write a C program to enter marks of five subjects and calculate total, average and percentage.
16. Write a C program to enter P, T, R and calculate Simple Interest.
17. Write a C program to enter P, T, R and calculate Compound Interest.

2. if else to practice for beginners.

18. Write a C program to find maximum between two numbers.
19. Write a C program to find maximum between three numbers.
20. Write a C program to check whether a number is negative, positive or zero.
21. Write a C program to check whether a number is divisible by 5 and 11 or not.
22. Write a C program to check whether a number is even or odd.
23. Write a C program to check whether a year is leap year or not.
24. Write a C program to check whether a character is alphabet or not.
25. Write a C program to input any alphabet and check whether it is vowel or consonant.
26. Write a C program to input any character and check whether it is alphabet, digit or special character.
27. Write a C program to check whether a character is uppercase or lowercase alphabet.
28. Write a C program to input week number and print week day.
29. Write a C program to input month number and print number of days in that month.
30. Write a C program to count total number of notes in given amount.
31. Write a C program to input angles of a triangle and check whether triangle is valid or not.

32. Write a C program to input all sides of a triangle and check whether triangle is valid or not.
33. Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle.
34. Write a C program to find all roots of a quadratic equation.
35. Write a C program to calculate profit or loss.
36. Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:
    Percentage >= 90% : Grade A
    Percentage >= 80% : Grade B
    Percentage >= 70% : Grade C
    Percentage >= 60% : Grade D
    Percentage >= 40% : Grade E
    Percentage < 40% : Grade F
37. Write a C program to input basic salary of an employee and calculate its Gross salary according to following:
    Basic Salary <= 10000 : HRA = 20%, DA = 80%
    Basic Salary <= 20000 : HRA = 25%, DA = 90%
    Basic Salary > 20000 : HRA = 30%, DA = 95%
38. Write a C program to input electricity unit charges and calculate total electricity bill according to the given condition:
    For first 50 units Rs. 0.50/unit
    For next 100 units Rs. 0.75/unit
    For next 100 units Rs. 1.20/unit
    For unit above 250 Rs. 1.50/unit
    An additional surcharge of 20% is added to the bill

3. Here is a set of basic programming problems based on switch case to practice.

39. Write a C program to print day of week name using switch case.
40. Write a C program print total number of days in a month using switch case.
41. Write a C program to check whether an alphabet is vowel or consonant using switch case.
42. Write a C program to find maximum between two numbers using switch case.
43. Write a C program to check whether a number is even or odd using switch case.
44. Write a C program to find roots of a quadratic equation using switch case.
45. Write a C program to create Simple Calculator using switch case.

4. Here is a list of programming problems based on conditional operator to practice for beginners.

46. Write a C program to find maximum between two numbers using conditional/ternary operator.

47. Write a C program to find maximum between three numbers using conditional/ternary operator.
48. Write a C program to check whether a number is even or odd using conditional/ternary operator.
49. Write a C program to check whether year is leap year or not using conditional/ternary operator.
50. Write a C program to check whether character is an alphabet or not using conditional/ternary operator.

5. Here is a set of programming exercises based on loops to practice.

51. Write a C program to print all natural numbers from 1 to n. - using while loop
52. Write a C program to print all natural numbers in reverse (from n to 1). - using while loop
53. Write a C program to print all alphabets from a to z. - using while loop
54. Write a C program to print all even numbers between 1 to 100. - using while loop
55. Write a C program to print all odd number between 1 to 100.
56. Write a C program to find sum of all natural numbers between 1 to n.
57. Write a C program to find sum of all even numbers between 1 to n.
58. Write a C program to find sum of all odd numbers between 1 to n.
59. Write a C program to print multiplication table of any number.
60.
61. Write a C program to count number of digits in any number.
62. Write a C program to find first and last digit of any number.
63. Write a C program to find sum of first and last digit of any number.
64. Write a C program to swap first and last digits of any number.
65. Write a C program to calculate sum of digits of any number.
66. Write a C program to calculate product of digits of any number.
67. Write a C program to enter any number and print its reverse.
68. Write a C program to enter any number and check whether the number is palindrome or not.
69. Write a C program to find frequency of each digit in a given integer.
70.
71. Write a C program to enter any number and print it in words.
72. Write a C program to print all ASCII character with their values.
73. Write a C program to find power of any number using for loop.
74. Write a C program to enter any number and print all factors of the number.
75. Write a C program to enter any number and calculate its factorial.
76. Write a C program to find HCF (GCD) of two numbers.
77. Write a C program to find LCM of two numbers.
78.
79. Write a C program to check whether a number is Prime number or not.
80. Write a C program to print all Prime numbers between 1 to n.
81. Write a C program to find sum of all prime numbers between 1 to n.
82. Write a C program to enter any number and print its prime factors.
83. Write a C program to check whether a number is Armstrong number or not.
84. Write a C program to print all Armstrong numbers between 1 to n.
85. Write a C program to check whether a number is Perfect number or not.
86. Write a C program to print all Perfect numbers between 1 to n.
87. Write a C program to check whether a number is Strong number or not.

88. Write a C program to print all Strong numbers between 1 to n.
89. Write a C program to print Fibonacci series up to n terms.
90.
91. Write a C program to find one's complement of a binary number.
92. Write a C program to find two's complement of a binary number.
93. Write a C program to convert Binary to Octal number system.
94. Write a C program to convert Binary to Decimal number system.
95. Write a C program to convert Binary to Hexadecimal number system.
96. Write a C program to convert Octal to Binary number system.
97. Write a C program to convert Octal to Decimal number system.
98. Write a C program to convert Octal to Hexadecimal number system.
99. Write a C program to convert Decimal to Binary number system.
100. Write a C program to convert Decimal to Octal number system.
101. Write a C program to convert Decimal to Hexadecimal number system.
102. Write a C program to convert Hexadecimal to Binary number system.
103. Write a C program to convert Hexadecimal to Octal number system.
104. Write a C program to convert Hexadecimal to Decimal number system.
105.
106. Write a C program to print Pascal triangle upto n rows.
107. Star pattern programs - Write a C program to print the given star patterns.
108. Number pattern programs - Write a C program to print the given number patterns.

6. Below is a set of programming exercises that can be used by a beginner or an intermediate programmer to master their skills on bitwise operator.

109. Write a C program to check Least Significant Bit (LSB) of a number is set or not.
110. Write a C program to check Most Significant Bit (MSB) of a number is set or not.
111. Write a C program to get nth bit of a number.
112. Write a C program to set nth bit of a number.
113. Write a C program to clear nth bit of a number.
114. Write a C program to toggle nth bit of a number.
115. Write a C program to get highest set bit of a number.
116. Write a C program to get lowest set bit of a number.
117. Write a C program to count trailing zeros in a binary number.
118. Write a C program to count leading zeros in a binary number.
119. Write a C program to flip bits of a binary number using bitwise operator.
120. Write a C program to total number of zeros and ones in a binary number.
121. Write a C program to convert decimal to binary number system using bitwise operator.
122. Write a C program to swap two numbers using bitwise operator.
123. Write a C program to check whether a number is even or odd using bitwise operator.

7. Number pattern programs in C

Number patterns are logical programs that basically are used to enhance your logical thinking abilities. These patterns are patterns created by numbers and are similar to star patterns. Enhance your logical thinking abilities by coding these patterns. Here is a list of basic, advanced, tricky and popular number patterns with logic and explanation. Learn how to print all these number pattern programs in C. Practice more and more of it to enhance your logical thinking.

7.1. Square number patterns

124. Number pattern 1

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

```
//pattern 01
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=0;j<=4;j++)
      printf("1\t");
      printf("\n");
   }
}
```

125. Number pattern 2

```
0 0 0 0 0
1 1 1 1 1
0 0 0 0 0
1 1 1 1 1
0 0 0 0 0
```

```
//pattern 02
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=0;j<=4;j++)
      if(j%2==0)
      printf("0\t");
      else
      printf("1\t");
      printf("\n");
   }
}
```

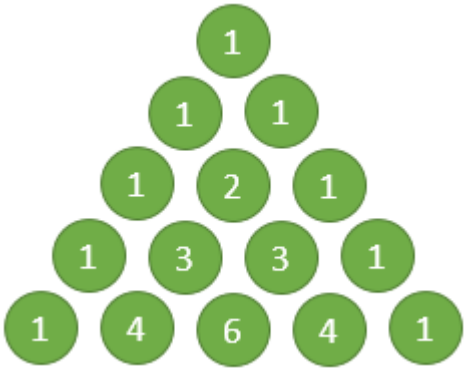
126. Number pattern 3

```
0 1 0 1 0
0 1 0 1 0
0 1 0 1 0
0 1 0 1 0
0 1 0 1 0
```

```
//pattern 127
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=5;i++)
   {
      for(j=0;j<=5;j++)
      {
         if
(i==0||i==5|j==0||j==5)
         {
            printf("1\t");
         }
         else
         {
            printf("0\t");
         }
      }
      printf("\n");
   }
}
```

127. Number pattern 4

```
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
```




```
//pattern 141
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("how many rows of dercimental
square you want:");
   scanf("%d",&num);
   for(i=num;i>=1;i--)
   {
      for(j=num;j>=i;j--)
      {
         printf("%d\t",j);
      }
      for(j=2;j<=i;j++)
      {
         printf("%d\t",i);
      }
      for(j=2;j<=i;j++)
      {
         printf("%d\t",i);
      }
      for(j=i+1;j<=num;j++)
      {
         printf("%d\t",j);
      }
      printf("\n");
   }
   for(i=2;i<=num;i++)
   {
      for(j=num;j>=i;j--)
      {
         printf("%d\t",j);
      }
      for(j=2;j<=i;j++)
      {
         printf("%d\t",i);
      }
      for(j=2;j<=i;j++)
      {
         printf("%d\t",i);
      }
      for(j=i+1;j<=num;j++)
      {
         printf("%d\t",j);
      }

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

```
1 1 1 1 1
```

128. Number pattern 5

```
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
0 1 0 1 0
1 0 1 0 1
```

129. Number pattern 6

```
1 1 0 1 1
1 1 0 1 1
0 0 0 0 0
1 1 0 1 1
1 1 0 1 1
```

130. Number pattern 7

```
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1
```

131. Number pattern 8

```
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0
```

132. Number pattern 9

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

133. Number pattern 10

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

134. Number pattern 11

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

135. Number pattern 12

```
1   2   3   4   5
6   7   8   9   10
11  12  13  14  15
16  17  18  19  20
```

```c
//pattern 10
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=1;j<=5;j++)
      printf("%d\t",j);
      printf("\n");
   }
}
```

```c
//pattern 128
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=0;j<=4;j++)
      {
         if
(i==0||i==4||j==0||j==4||i==1||i
==3||j==3||j==1)
         {
            printf("1\t");
         }
         else
         {
            printf("0\t");
         }
      }
      printf("\n");
   }
}
```

```c
//pattern 171
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter how many rows of
v pattern you want:");
   scanf("%d",&num);
   for(i=1;i<=num;i++)
   {
      for(j=1;j<=i;j++)
      {
         printf("%d\t",j);
      }
      for(k=num;k>i;k--)
      {
         printf("^\t");
      }
       for(j=num-1;j>=i;j--)
      {
         printf("^\t");
      }
      for(k=i;k>=1;k--)
      {
         printf("%d\t",k);
      }
      printf("\n");
   }
}
```

```c
//pattern 140
#include<stdio.h>
void main()
{
   int i,j,k=5;
   for(i=1;i<=5;i++)
   {
      for(j=i;j>=1;j--)
      {
         printf("%d\t",j);
      }
      for(j=2;j<=k;j++)
      {
         printf("%d\t",j);
      }
      printf("\n");
      k--;
   }
}
```

```c
//pattern 131
#include<stdio.h>
void main()
{
   int i,j,k=1,l=1,m,num;
   printf("enter number of rows you want: ");
   scanf("%d",&num);
   m=num;
   for(i=1;i<=num;i++)
   {
      for(j=1;j<=num;j++)
      {
         if(i==k&&j==k||i==l&&j==m)
         {
            printf("1\t");
         }
         else
         {
            printf("0\t");
         }
      }
      printf("\n");
      k++;
      l++;
      m--;
   }
}
```

```c
//pattern 09
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=0;j<=4;j++)
      printf("%d\t",i);
      printf("\n");
   }
}
```

```c
//pattern 11
#include<stdio.h>
void main()
{
   int i,j,k=5;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=k;j++)
      printf("%d\t",j);
      printf("\n");
      k++;
   }
}
```

```c
//pattern 12
#include<stdio.h>
void main()
{
   int i,j,k=1;
   for(i=5;i<=25;i=i+5)
   {
      for(j=k;j<=i;j++)
      printf("%d\t",j);
      printf("\n");
      k=k+5;
   }
}
```

```
21  22  23  24  25
```

136. Number pattern 13

```
55555
54444
54333
54322
54321
```

137. Number pattern 14

```
12345
23455
34555
45555
55555
```

138. Number pattern 15

```
12345
23451
34521
45321
54321
```

139. Number pattern 16

```
12345
21234
32123
43212
54321
```

140. Number pattern 17

```
5  5  5  5  5  5  5  5  5
5  4  4  4  4  4  4  4  5
5  4  3  3  3  3  3  4  5
5  4  3  2  2  2  3  4  5
5  4  3  2  1  2  3  4  5
5  4  3  2  2  2  3  4  5
5  4  3  3  3  3  3  4  5
5  4  4  4  4  4  4  4  5
5  5  5  5  5  5  5  5  5
```

141. Number pattern 18

```
1     2     3     4     5
16    17    18    19    6
15    24    25    20    7
14    23    22    21    8
13    12    11    10    9
```

142. Number pattern 19

```
//pattern 138
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=4;j++)
      {
         printf("%d",j);
      }
      for(k=1;k<=i;k++)
      {
         printf("5");
      }
      printf("\n");
   }
}
```

```
//pattern 139
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=5;j++)
      {
         printf("%d",j);
      }
      for(k=i-1;k>=1;k--)
      {
         printf("%d",k);
      }
      printf("\n");
   }
}
```

```
//pattern 137
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=0;j<=4;j++)
      {
         if(i==0||j==0)
         {
         printf("5");
         }
         else if(i==1||j==1)
         {
            printf("4");
         }
         else if(i==2||j==2)
         {
            printf("3");
         }
         else if(i==3||j==3)
         {
            printf("2");
         }
         else if(i==4||j==4)
         {
            printf("1");
         }
      }
      printf("\n");
   }
}
```

```
//pattern 141
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=8;i++)
   {
      for(j=0;j<=8;j++)
      {
         if(i==0||i==8||j==0||j==8)
         {
            printf("5\t");
         }
         else if(i==7||j==7||i==1||j==1)
         {
            printf("4\t");
         }
         else if(i==6||j==6||i==2||j==2)
         {
            printf("3\t");
         }
         else if(i==5||j==5||i==3||j==3)
         {
            printf("2\t");
         }
         else if(i==4||j==4||i==4||j==4)
         {
            printf("1\t");
         }

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

7.2. Triangle easy number patterns

```
1
22
333
```

```
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=5;i++)
   {
      for(j=1;j<=i;j++)
      {
         printf("%d",i);
      }
      printf("\n");
   }

}
```

```
4 4 4 4
5 5 5 5 5
```

143. Number pattern 20

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

144. Number pattern 21

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

145. Number pattern 22

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

146. Number pattern 23

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

147. Number pattern 24

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

148. Number pattern 25

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

149. Number pattern 26

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

150. Number pattern 27

```
5
```

```
//pattern 20
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i<=5;i--)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d",i);
        }
        printf("\n");
    }
}
```

```
//pattern 21
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=5;i++)
    {
        for(j=5;j>=i;j--)
        {
            printf("%d",i);
        }
        printf("\n");
    }
}
```

```
//pattern 22
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i>=1;i--)
    {
        for(j=5;j>=i;j--)
        printf("%d",i);
        printf("\n");
    }
}
```

```
//pattern 23
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=5;i++)
    {
        for(j=1;j<=i;j++)
        printf("%d",j);
        printf("\n");
    }
}
```

```
//pattern 24
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i>=1;i--)
    {
        for(j=1;j<=i;j++)
        printf("%d",j);
        printf("\n");
    }
}
```

```
//pattern 25
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=5;i++)
    {
        for(j=i;j>=1;j--)
        printf("%d",j);
        printf("\n");
    }
}
```

```
//pattern 26
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i>=1;i--)
    {
        for(j=i;j>=1;j--)
        printf("%d",j);
        printf("\n");
    }
}
```

```
//pattern 27
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i>=1;i--)
    {
        for(j=5;j>=i;j--)
        printf("%d",j);
        printf("\n");
    }
}
```

```
5 4
5 4 3
5 4 3 2
5 4 3 2 1
```

151. Number pattern 28

```
//pattern 28
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=5;i++)
    {
        for(j=5;j>=i;j--)
        printf("%d",j);
        printf("\n");
    }
}
```

```
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

152. Number pattern 29

```
//pattern 29
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=5;i>=1;i--)
    {
        for(j=i;j<=5;j++)
        printf("%d",j);
        printf("\n");
    }
}
```

```
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5
```

153. Number pattern 30

```
//pattern 31
#include<stdio.h>
void main()
{
    int i,k=1;
    static int j=1;

    for(i=1;i<=5;i++)
    {
        for(j=i;j<=k;j++)
        {
        printf("%d",j);

        }
        printf("\n");
        k=k+2;
    }

}
```

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

154. Number pattern 31

```
//pattern 30
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=5;i++)
    {
        for(j=i;j<=5;j++)
        printf("%d",j);
        printf("\n");
    }
}
```

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

155. Number pattern 32

```
//pattern 32
#include<stdio.h>
void main()
{
    int i,k=9;
    static int j=1;
    for(i=5;i<=9;i--)
    {
        for(j=i;j<=k;j++)
        {
        printf("%d",j);
        }
        printf("\n");
        k=k-2;
    }
}
```

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

156. Number pattern 33

```
1 3 5 7 9
3 5 7 9
5 7 9
7 9
9
```

157. Number pattern 34

```
//pattern 132
#include<stdio.h>
void main()
{
    int i,j,k=1,l=1,m,num;
    printf("enter number of rows you want: ");
    scanf("%d",&num);
    m=num;
    for(i=1;i<=num;i++)
    {
        for(j=1;j<=num;j++)
        {
            if(i==1||i==num||j==1||j==num)
            {
                if
(i==1&&j==1||i==num&&j==num||i==num&&j==1||i==1&&j==num)
            {
                printf("0\t");
                continue;
            }
                printf("1\t");
            }
            else
            {
                printf("0\t");
            }
        }
        printf("\n");
    }
}
```

7.3. Triangle 0,1 easy patterns

```
1
1 0
```

```
1 0 1
1 0 1 0
1 0 1 0 1
```

158. Number pattern 35

```
1
0 0
1 1 1
0 0 0 0
1 1 1 1 1
```

159. Number pattern 36

```
1
0 1
0 1 0
1 0 1 0
1 0 1 0 1
```

160. Number pattern 37

```
1
1 1
1 0 1
1 0 0 1
1 1 1 1 1
```

161. Number pattern 38

7.4. Triangle hard number patterns

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

162. Number pattern 39

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

163. Number pattern 40

```
1
1 3 1
1 3 5 3 1
1 3 5 7 5 3 1
1 3 5 7 9 7 5 3 1
```

164. Number pattern 41

```
2
2 4 2
2 4 6 4 2
2 4 6 8 6 4 2
2 4 6 8 1 0 8 6 4 2
```

```
//pattern 168
#include<stdio.h>
void main()
{
   int i,j,k=1,num;
   printf("enter number of rows you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("%d\t",k);
         k++;
      }
      printf("\n");
   }
}
```

```
//pattern 39
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=9;i=i+2)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }

}
```

```
//pattern 157
#include<stdio.h>
void main()
{
   int i,j,k=1;
   for(i=0;i<=5;i++)
   {
      for(j=k;j<=9;j=j+2)
      {
         printf("%d\t",j);
      }
      printf("\n");
      k=k+2;
   }
}
```

```
//pattern 36
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=5;j++)
      {
         printf(" ");
      }
      for(k=1;k<=i;k++){
         printf("%d",k);
      }
      for(j=i-1;j>=1;j--)
      {
         printf("%d",j);
      }
      printf("\n");
   }
}
```

```
 1
 121
 12321
 1234321
 123454321
```

```
//pattern 36
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=1;j<=i;j++)
      {
         if(i%2==0)
         printf("0\t");
         else
         printf("1\t");
      }
      printf("\n");
   }
}
```

```
//pattern 35
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=1;j<=i;j++)
      {
         if(j%2==0)
         printf("0\t");
         else
         printf("1\t");
      }
      printf("\n");
   }
}
```

165. Number pattern 42

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

166. Number pattern 43

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

167. Number pattern 44

```
1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
```

168. Number pattern 45

```
1
2 1
1 2 3
4 3 2 1
1 2 3 4 5
```

169. Number pattern 46

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

170. Number pattern 47

```
1             1
1 2         2 1
1 2 3     3 2 1
1 2 3 4 4 3 2 1
1 2 3 4 5 5 4 3 2 1
```

171. Number pattern 48

```
1
2   6
3   7   10
4   8   11  13
5   9   12  14  15
```

172. Number pattern 49

```
1
2   4
7   11  16
22  29  37  46
```

```
//pattern 36
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=5;j++)
      {
         printf("^");
      }
      for(k=1;k<=i;k++){
         printf("%d",k);
      }

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

```
/*
^^^^^1
^^^^12
^^^123
^^1234
^12345

*/
```

```
//pattern 36
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j>=1;j--)
      {
         printf("%d",j);
      }
      for(k=4;k>=i;k--){
         printf("^");
      }
      for(j=i;j<=4;j++)
      {
         printf("^");
      }
      for(k=1;k<=i;k++){
         printf("%d",k);
      }
       printf("\n");
   }
}
```

```
1^^^^^^^^1
21^^^^^^12
321^^^^123
4321^^1234
5432112345
```

```
/*
//pattern 36
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j>=1;j--)
      {
         printf("%d",j);
      }
      for(k=4;k>=i;k--){
         printf("^");
      }
       printf("\n");
   }
```

```
}

1^^^^
21^^^
321^^
4321^
54321
*/
```

```
//pattern(diamond)
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=5;j++)
      {
         printf("^");
      }
      for(k=1;k<=i;k++){
         printf("%d",k);
      }
      for(j=i-1;j>=1;j--)
      {
         printf("%d",j);
      }
      printf("\n");
   }
   for(i=5;i>=1;i--)
   {
      for(k=5;k>=i;k--)
      printf("^");
      for(j=1;j<=i-1;j++)
      printf("%d",j);
      for(j=i;j>=1;j--)
         printf("%d",j);
         for(k=5;k>=i;k--)
         printf("^");
      printf("\n");
   }
}
```

```
^^^^^1
^^^^121
^^^12321
^^1234321
^123454321
^123454321^
^^1234321^^
^^^12321^^^
^^^^121^^^^
^^^^^1^^^^^
```

```
56  67  79  92  106
```

173. Number pattern 50

```
1
3   2
4   5   6
10  9   8   7
11  12  13  14  15
```

174. Number pattern 51

```
1
22
333
2222
11111
```

175. Number pattern 52

```
N = 12345

12345
1234
123
12
1
```

176. Number pattern 53

```
N = 12345

12345
2345
345
45
5
```

177. Number pattern 54

```
//pattern 175
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter the number of rows you want:");
   scanf("%d",&num);
  if(num%2==0)
   {
      k=num/2;
   }
   else
   {
      k=(num+1)/2;
   }

   for(i=1;i<=num;i++)
   {
      for(j=1;j<=i;j++)
      {
         if(i<=k)
         {
            printf("%d",i);
         }
         else
         {
            printf("%d",k);
         }
      }
      if(i>=k)
      {
         k--;
      }
      printf("\n");
   }
}
```

```
//pattern 161
#include<stdio.h>
void main()
{
   int i,j,k=0,num;
   printf("enter how many rows you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         if(j==0||i==num||i==k&&j==k)
         {
         printf("1");
         }
         else
         {
            printf("0");
         }
      }
      printf("\n");
      k++;
   }
}
```

```
//pattern 53
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=i;j<=5;j++)
      printf("%d",j);
      printf("\n");
   }
}
```

```
//pattern 52
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=5;i>=1;i--)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }
}
```

7.5. Diamond number patterns

```
1
12
123
1234
12345
1234
123
12
1
```

178. Number pattern 55

```
1
123
12345
1234567
123456789
```

```
//pattern 54
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=5;i++)
   {
      for(j=1;j<=i;j++)
      printf("%d\t",j);
      printf("\n");
   }
   for(i=4;i>=1;i--)
   {
      for(j=1;j<=i;j++)
      printf("%d\t",j);
      printf("\n");
   }
}
```

```
//pattern 55
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=9;i=i+2)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }
}
```

```
//pattern 55(total diamond )
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=1;i<=9;i=i+2)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }
for(i=7;i>=1;i=i-2)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }
}
```

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

179. Number pattern 56

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1
```

180. Number pattern 57

```
//pattern 55(lower triangle)
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=7;i>=1;i=i-2)
   {
      for(j=1;j<=i;j++)
      printf("%d",j);
      printf("\n");
   }
}
```

```
//pattern 130
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter number of
rows:");
   scanf("%d",&num);
   if(num%2==0)
   {
      k=num/2;
   }
   else
   {
      k=(num+1)/2;
   }
   for(i=1;i<=num;i++)
   {
      for(j=1;j<=num;j++)
      {
         if(i==k||j==k)
         {
            printf("0\t");
         }
         else
         {
            printf("1\t");
         }
      }
      printf("\n");
   }
}
```

7.6. Diamond number pattern with star border

```
*
* 1 *
* 1 2 1 *
* 1 2 3 2 1 *
* 1 2 3 4 3 2 1 *
* 1 2 3 4 5 4 3 2 1 *
* 1 2 3 4 3 2 1 *
* 1 2 3 2 1 *
* 1 2 1 *
* 1 *
*
```

181. Number pattern 58

```
//pattern 129
#include<stdio.h>
void main()
{
   int i,j,k=1,l=1,m,num,mid;
   printf("enter number of rows you want: ");
   scanf("%d",&num);
   m=num;
   if(num%2==0)
   {
      mid=num/2;
   }
else
   {
      mid=(num+1)/2;
   }
   for(i=1;i<=num;i++)
   {
      for(j=1;j<=num;j++)
      {
         if
(i==k&&j==k||i==l&&j==m||i==1&&j==mid||i==num
&&j==mid||j==1&&i==mid||j==num&&i==mid)
         {
            printf("1\t");
         }

         else
         {
            printf("0\t");
         }
      }
   printf("\n");
   k++;
   l++;
   m--;
   }
}
```

7.7. Tricky number pattern

```
1           1
  2       2
    3   3
      4 4
        5
      4 4
    3   3
  2       2
1           1
```

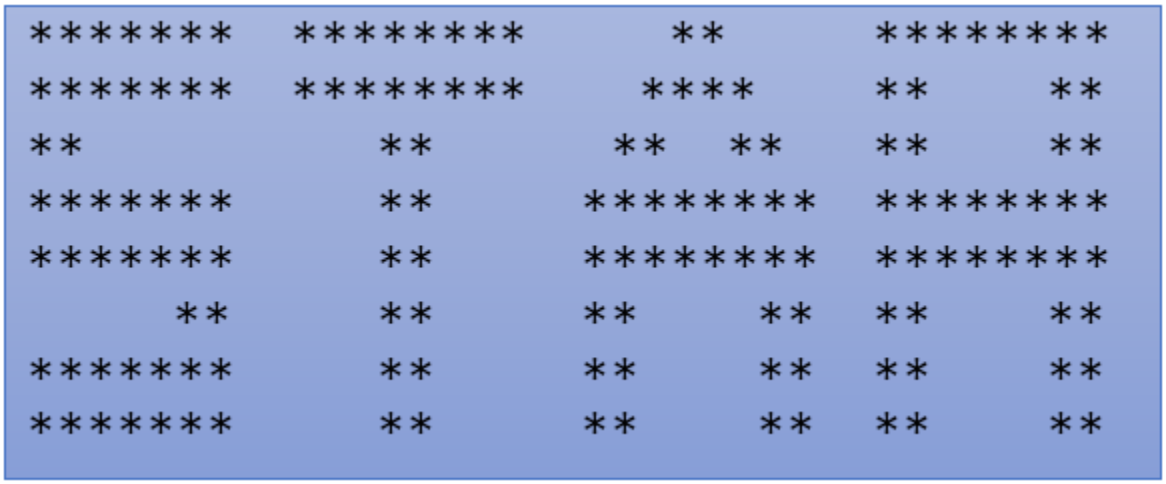
182. Number pattern 59

Happy coding ;)

7.8. Star patterns in C programming

Star patterns are a series of '*' or any other character that are used to create some patterns or any geometrical shape such as - square, triangle(Pyramid), rhombus, heart etc. These patterns are often prescribed by many programming books and are best for practicing loops in programming and to enhance logical thinking capability. Before printing any star pattern you must have knowledge of loopsand basic of pattern printing (Logic to print star patterns and shape).

Below is a list of easy and hard star patterns in C programming with explanation. Practice as much as you can to increase your logical thinking.



7.9. List of various star pattern series and solution in C programming:

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

183. Square

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

184. Hollow Square

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

```
//pattern 185
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=5;i++)
   {
      for(j=5;j>=i;j--)
      {
         printf("^");
      }
      for(j=0;j<=5;j++)
      {
      printf("*");
      }
      for(k=0;k<=i;k++)
      {
         printf("^");
      }
      printf("\n");
   }
}
```

```
//pattern 187
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=5;i++)
   {
      for(k=0;k<=i;k++)
      {
         printf("^");
      }
      for(j=0;j<=5;j++)
      {
      printf("*");
      }
      for(j=5;j>=i;j--)
      {
         printf("^");
      }
      printf("\n");
   }
}
```

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

185. Rhombus

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

186. Hollow Rhombus

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

187. Mirrored Rhombus

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

188. Hollow mirrored Rhombus

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

189. Right triangle

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

190. Hollow right triangle

```
//pattern 180
#include<stdio.h>
void main()
{
    int i,j,k=4,l=3;
    for(i=1;i<=5;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d\t",j);
        }
        for(j=i-1;j>=1;j--)
        {
            printf("%d\t",j);
        }
        printf("\n");
    }
    for(i=1;i<=5;i++)
    {
        for(j=1;j<=k;j++)
        {
            printf("%d\t",j);
        }
        for(j=l;j>=1;j--)
        {
            printf("%d\t",j);
        }

        printf("\n");
        k--;
        l--;
    }
}
```

```
//pattern 181
#include<stdio.h>
void main()
{
    int i,j,k=4,l=3;
    for(i=1;i<=5;i++)
    {
        for(j=0;j<=i;j++)
        {
            if(j==0)
            {
                printf("*\t");
                continue;
            }
            printf("%d\t",j);
        }
        for(j=i-1;j>=1;j--)
        {
            printf("%d\t",j);
        }
        printf("*");
        printf("\n");
    }
    for(i=1;i<=5;i++)
    {
        for(j=0;j<=k;j++)
        {
            if(j==0)
            {
                printf("*\t");
                continue;
            }
            printf("%d\t",j);
        }
        for(j=l;j>=1;j--)
        {
            printf("%d\t",j);
        }
        printf("*\t");
        printf("\n");
        k--;
        l--;
    }
}
```

```
//pattern 190
#include<stdio.h>
void main()
{
    int i,j,k=0;
    for(i=0;i<=10;i++)
    {
        for(j=0;j<=10;j++)
        {
            if(i==10||j==0||i==k&&j==k)
            {
                printf("*\t");
            }
            else
            printf("^\t");
        }
        printf("\n");
        k++;
    }
}
```

```
//pattern 189
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=0;i<=5;i++)
    {
        for(j=0;j<=i;j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

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

191. Mirrored right triangle

```
//pattern 191
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=0;i<=6;i++)
    {
        for(j=6;j>=i;j--)
        printf("^");
        for(k=0;k<=i;k++)
        printf("*");
        printf("\n");
    }
}
```

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

192. Hollow mirrored right triangle

```
//pattern 192
#include<stdio.h>
void main()
{
    int i,j,k=0,l=5;
    for(i=0;i<=5;i++)
    {
        for(j=0;j<=5;j++)
        {
            if(i==5||j==5||i==k&&j==l)
            {
                printf("*\t");
            }
                else
                {
                printf("^\t");
                }
        }
        printf("\n");
        k++;
        l--;
    }
}
```

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

193. Inverted right triangle

```
//pattern 193
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=0;i<=5;i++)
    {
    for(j=5;j>=i;j--)
    {
    printf("*");
    }
    printf("\n");
    }
}
```

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

194. Hollow inverted right triangle

```
//pattern 194
#include<stdio.h>
void main()
{
    int i,j,k=0,l=8;
    for(i=0;i<=8;i++)
    {
        for(j=0;j<=8;j++)
        {
            if(i==0||j==0||i==k&&j==l)
            {
                printf("*\t");
            }
                else
                {
                printf("^\t");
                }
        }
        printf("\n");
        k++;
        l--;
    }
}
```

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

195. Inverted mirrored right triangle

```
//pattern 195
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=0;i<=5;i++)
    {
        for(j=0;j<=i;j++)
        {
            printf("^");
        }
        for(k=5;k>=i;k--)
            {
                printf("*");
            }printf("\n");
    }
}
```

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

```
//pattern 196
#include<stdio.h>
void main()
{
    int i,j,k=0;
    for(i=0;i<=5;i++)
    {
        for(j=0;j<=5;j++)
        {
            if(i==0||j==5||i==k&&j==k)
            {
                printf("*\t");
            }
            else
            {
                printf("^\t");
            }
        }
        printf("\n");
        k++;
    }
```

```
    *  *
     **
      *
```

196. Hollow inverted mirrored right triangle

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

197. Pyramid (Equilateral triangle)

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

198. Hollow Pyramid

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

199. Inverted Pyramid

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

200. Hollow inverted pyramid

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

```
//pattern 197
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter number of rows pyramid you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=num;j>=i;j--)
      {
         printf("^");
      }
      for(k=0;k<=i;k++)
         {
            printf("*");
         }
         for(j=1;j<=i;j++)
      {
         printf("*");
      }
      for(k=num;k>=i;k--)
      {
         printf("^");
      }
         printf("\n");
   }
}
```

```
//pattern 198
#include<stdio.h>
void main()
{
   int i,j,k=0,l,num;
   printf("enter how many rows of
pyramid you want:");
   scanf("%d",&num);
   l=num;
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=num;j++)
      {
         if(i==num||i==k&&j==l)
         {
         printf("*");
         }
         else
         {
            printf("^");
         }
      }
      for(j=0;j<=num;j++)
      {
         if(i==num||i==k&&j==k)
         {
            if(i==0&&j==0)
            {
               break;
            }
            printf("*");
         }
         else
         {
            printf("^");
         }
      }
      printf("\n");
      l--;
      k++;
   }
}
```

```
//pattern 199
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter how many rows of
inverted pyramid you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("^");
      }
      for(k=num;k>=i;k--)
      {
         printf("*");
      }
      for(j=num;j>i;j--)
      {
         printf("*");
      }
      for(k=0;k<=i;k++)
      {
         printf("^");
      }
      printf("\n");
   }
}
```

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

```
//pattern 200
#include<stdio.h>
void main()
{
   int i,j,m=0,l,k=0,num;
   printf("enter how many rows
of inverted hollow pyramid you
want:");
   scanf("%d",&num);
   l=num-1;
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=num;j++)
      {
         if(i==0||i==m&&j==m)
         {
         printf("*");
         }
         else
         {
            printf(" ");
         }
      }
       m++;
       for(j=0;j<=num;j++)
      {
         if(i==0||i==k&&j==l)
         {
         printf("*");
         }
         else
         {
            printf(" ");
         }
      }
       k++;
      l--;
      printf("\n");
   }
}
```

201. Half diamond

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

```
//pattern 201
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter how many colums
of half diamond you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("*");
      }
      printf("\n");
   }
   for(i=0;i<=num;i++)
   {
      for(j=num-1;j>=i;j--)
      {
         printf("*");
      }
      printf("\n");
   }
}
```

202. Mirrored half diamond

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

```
//pattern 203
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("how many rows of diamond
you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=num;j>=i;j--)
      {
         printf("^");
      }
      for(k=0;k<=i;k++)
      {
         printf("*");
      }
      for(j=1;j<=i;j++)
      {
         printf("*");
      }
      printf("\n");
   }
    for(i=1;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("^");
      }
      for(k=num;k>=i;k--)
      {
         printf("*");
      }
      for(j=num-1;j>=i;j--)
      {
         printf("*");
      }
      printf("\n");
   }
```

203. Diamond

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

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

204. Hollow diamond

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

205. Right Arrow

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

206. Left arrow

```
    +
    +
    +
    +
+++++++++
    +
    +
    +
    +
```

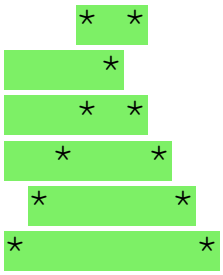
207. Plus Star pattern

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

```
//pattern 202
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("how many rows of diamond you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=num;j>=i;j--)
      {
         printf("^");
      }
      for(k=0;k<=i;k++)
      {
         printf("*");
      }
      printf("\n");
   }
    for(i=1;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("^");
      }
      for(k=num;k>=i;k--)
      {
         printf("*");
      }
      printf("\n");
   }
}
```

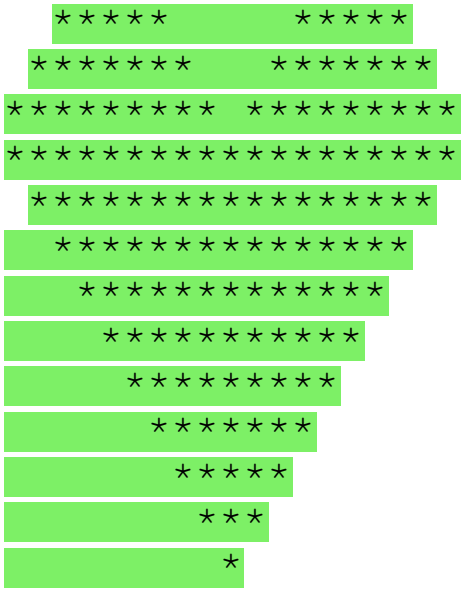
```
//pattern 207
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=8;i++)
   {
      for(j=0;j<=8;j++)
      {
         if(i==4||j==4)
         {
         printf("+\t");
         }
         else
         {
            printf("^\t");
         }
      }
      printf("\n");
   }
}
```

```
//pattern 204
#include<stdio.h>
void main()
{
   int i,j,k,num;
   printf("enter how many rows of hollow diamond you want:");
   scanf("%d",&num);
   for(i=0;i<=num;i++)
   {
      for(j=num;j>=i;j--)
      {
         printf("*");
      }
      for(k=0;k<=i;k++)
      {
         printf("^");
      }
      for(j=1;j<=i;j++)
      {
         printf("^");
      }
      for(k=num;k>=i;k--)
      {
         printf("*");
      }
      printf("\n");
   }
   for(i=0;i<=num;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("*");
      }
      for(k=num;k>=i;k--)
      {
         printf("^");
      }
      for(j=num-1;j>=i;j--)
      {
         printf("^");
      }
      for(k=0;k<=i;k++)
      {
         printf("*");
      }
      printf("\n");
   }
}
```



208. X Star pattern

209. Eight (8) Star pattern



210. 28. Heart Star pattern 1

```
  *****         *****
 *******       *******
*********     *********
*****Codeforwin****
```

```
//pattern 208
#include<stdio.h>
void main()
{
   int i,j,k=0,l=5,a=0,b=5;
   for(i=0;i<=5;i++)
   {
      for(j=0;j<=5;j++)
      {
         if(i==k&&j==k)
         {
         printf("*");
         }
         else
         {
            printf("^");
         }
      }
      for(j=0;j<=5;j++)
      {
         if(i==k&&j==l)
      {
         printf("*");
      }
      else
      {
         printf("^");
      }
      }
      printf("\n");
      k++;
      l--;
   }


 for(i=0;i<=4;i++)
  {
     for(j=0;j<=5;j++)
     {
        if(i==a&&j==b)
        {
           if(a==0&&b==5)
           {
              break;
           }
        printf("*");
        }
           else
           {
           printf("^");
           }
     }
      for(j=0;j<=5;j++)
     {
        if(i==a&&j==a)
        {

        printf("*");
        }
        else
        {
           printf("^");
        }
     }
      a++;
      b--;
      printf("\n");
  }

}
```

```
//pattern 210(heart symbol)
 #include<stdio.h>
 void main()
 {
    int i,j,k,num;
    printf("enter heart size(1-10):");
    scanf("%d",&num);
     for(i=0;i<=num;i++)
  {
     for(j=num;j>=i;j--)
     {
        printf("^");
     }
     for(k=0;k<=i;k++)
     {
        printf("*");
     }
     for(j=1;j<=i;j++)
     {
        printf("*");
     }
     for(j=num;j>=i;j--)
     {
        printf("^");
     }
     for(j=num;j>=i;j--)
     {
        printf("^");
     }
     for(k=0;k<=i;k++)
     {
        printf("*");
     }
     for(j=1;j<=i;j++)
     {
        printf("*");
     }
     for(j=num;j>=i;j--)
     {
        printf("^");
     }
     printf("\n");
  }
  num=num*2.3;
  for(i=0;i<=num;i++)
  {
     for(j=0;j<=i;j++)
     {
        printf("^");
     }
     for(k=num;k>=i;k--)
     {
        printf("*");
     }
     for(j=num;j>i;j--)
     {
        printf("*");
     }
     for(k=0;k<=i;k++)
     {
        printf("^");
     }
     printf("\n");
  }
 }
```

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

211. 29. Heart Star pattern 2

```
//pattern 206
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=5;j>=i;j--)
         {
            printf("^\t");
         }

      for(j=5;j>=i;j--)
         {
            printf("*");
         }
      printf("\n");
   }
   for(i=0;i<=5;i++)
      {
         for(j=0;j<=i;j++)
      {
         printf("^\t");
      }

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

```
//pattern 205
#include<stdio.h>
void main()
{
   int i,j,k;
   for(i=0;i<=4;i++)
   {
      for(j=0;j<=i;j++)
      {
         printf("^\t");
      }
      for(j=5;j>=i;j--)
         {
            printf("*");
         }
      printf("\n");
   }
   for(i=0;i<=5;i++)
      {
         for(j=5;j>=i;j--)
         {
            printf("^\t");
         }
         for(j=0;j<=i;j++)
         {
            printf("*");
         }
         printf("\n");
      }
}
```

8. Array and matrix to practice for beginners:

212. Write a C program to read and print elements of array. - using recursion.
213. Write a C program to print all negative elements in an array.
214. Write a C program to find sum of all array elements. - using recursion.
215. Write a C program to find maximum and minimum element in an array. - using recursion.
216. Write a C program to find second largest element in an array.
217. Write a C program to count total number of even and odd elements in an array.
218. Write a C program to count total number of negative elements in an array.
219. Write a C program to copy all elements from an array to another array.
220. Write a C program to insert an element in an array.
221. Write a C program to delete an element from an array at specified position.
222. Write a C program to print all unique elements in the array.
223. Write a C program to count total number of duplicate elements in an array.
224. Write a C program to delete all duplicate elements from an array.
225. Write a C program to count frequency of each element in an array.
226. Write a C program to merge two array to third array.
227. Write a C program to find reverse of an array.
228. Write a C program to put even and odd elements of array in two separate array.
229. Write a C program to search an element in an array.
230. Write a C program to sort array elements in ascending order.
231. Write a C program to sort array elements in descending order.
232. Write a C program to sort even and odd elements of array separately.
233. Write a C program to left rotate an array.
234. Write a C program to right rotate an array.
235.

236. Write a C program to add two matrices.
237. Write a C program to subtract two matrices.
238. Write a C program to perform Scalar matrix multiplication.
239. Write a C program to multiply two matrices.
240. Write a C program to check whether two matrices are equal or not.
241. Write a C program to find sum of main diagonal elements of a matrix.
242. Write a C program to find sum of minor diagonal elements of a matrix.
243. Write a C program to find sum of each row and column of a matrix.
244. Write a C program to interchange diagonals of a matrix.
245. Write a C program to find upper triangular matrix.
246. Write a C program to find lower triangular matrix.
247. Write a C program to find sum of upper triangular matrix.
248. Write a C program to find sum of lower triangular matrix.
249. Write a C program to find transpose of a matrix.
250. Write a C program to find determinant of a matrix.
251. Write a C program to check Identity matrix.
252. Write a C program to check Sparse matrix.
253. Write a C program to check Symmetric matrix.

9. Strings

Strings are basically array of characters that represent some textual data in a program. Here are basic string programs with detailed explanation that will help to enhance your string programming skills. These exercises can be practiced by anyone a beginner or an intermediate programmers.

254. Write a C program to find length of a string.
255. Write a C program to copy one string to another string.
256. Write a C program to concatenate two strings.
257. Write a C program to compare two strings.
258.
259. Write a C program to convert lowercase string to uppercase.
260. Write a C program to convert uppercase string to lowercase.
261. Write a C program to toggle case of each character of a string.
262.
263. Write a C program to find total number of alphabets, digits or special character in a string.
264. Write a C program to count total number of vowels and consonants in a string.
265. Write a C program to count total number of words in a string.
266. Write a C program to find reverse of a string.
267. Write a C program to check whether a string is palindrome or not.
268. Write a C program to reverse order of words in a given string.
269.

270. Write a C program to find first occurrence of a character in a given string.
271. Write a C program to find last occurrence of a character in a given string.
272. Write a C program to search all occurrences of a character in given string.
273. Write a C program to count occurrences of a character in given string.
274. Write a C program to find highest frequency character in a string.
275. Write a C program to find lowest frequency character in a string.
276. Write a C program to count frequency of each character in a string.
277. Write a C program to remove first occurrence of a character from string.
278. Write a C program to remove last occurrence of a character from string.
279. Write a C program to remove all occurrences of a character from string.
280. Write a C program to remove all repeated characters from a given string.
281. Write a C program to replace first occurrence of a character with another in a string.
282. Write a C program to replace last occurrence of a character with another in a string.
283. Write a C program to replace all occurrences of a character with another in a string.
284.
285. Write a C program to find first occurrence of a word in a given string.
286. Write a C program to find last occurrence of a word in a given string.
287. Write a C program to search all occurrences of a word in given string.
288. Write a C program to count occurrences of a word in a given string.
289. Write a C program to remove first occurrence of a word from string.
290. Write a C program to remove last occurrence of a word in given string.
291. Write a C program to remove all occurrence of a word in given string.
292.
293. Write a C program to trim leading white space characters in a string.
294. Write a C program to trim trailing white space characters in a string.
295. Write a C program to trim both leading and trailing white space characters in a string.
296. Write a C program to remove all extra blank spaces from a given string.

10. Functions

Functions in programming is a block of statements grouped together to perform some specific task. Functions provides modularity to our program which are easy to maintain, debug and understand. Functions in C programming can be classified in two category i.e. predefined or library functions and user defined functions. Below is a list of programming exercises based on functions and recursion to practice for beginners to advance their programming capabilities with functions. Here in this programming exercise we will mainly focus on user defined functions and recursion.

297. Write a C program to find cube of any number using function.

298. Write a C program to find diameter, circumference and area of circle using functions.
299. Write a C program to find maximum and minimum between two numbers using functions.
300. Write a C program to check whether a number is even or odd using functions.
301. Write a C program to check whether a number is prime, Armstrong or perfect number using functions.
302. Write a C program to find all prime numbers between given interval using functions.
303. Write a C program to print all strong numbers between given interval using functions.
304. Write a C program to print all armstrong numbers between given interval using functions.
305. Write a C program to print all perfect numbers between given interval using functions.
306.
307. Write a C program to find power of any number using recursion.
308. Write a C program to print all natural numbers between 1 to n using recursion.
309. Write a C program to print all even or odd numbers in given range using recursion.
310. Write a C program to find sum of all natural numbers between 1 to n using recursion.
311. Write a C program to find sum of all even or odd numbers in given range using recursion.
312. Write a C program to find reverse of any number using recursion.
313. Write a C program to check whether a number is palindrome or not using recursion.
314. Write a C program to find sum of digits of a given number using recursion.
315. Write a C program to find factorial of any number using recursion.
316. Write a C program to generate nth Fibonacci term using recursion.
317. Write a C program to find GCD (HCF) of two numbers using recursion.
318. Write a C program to find LCM of two numbers using recursion.
319.
320. Write a C program to display all array elements using recursion.
321. Write a C program to find sum of elements of array using recursion.
322. Write a C program to find maximum and minimum elements in array using recursion.