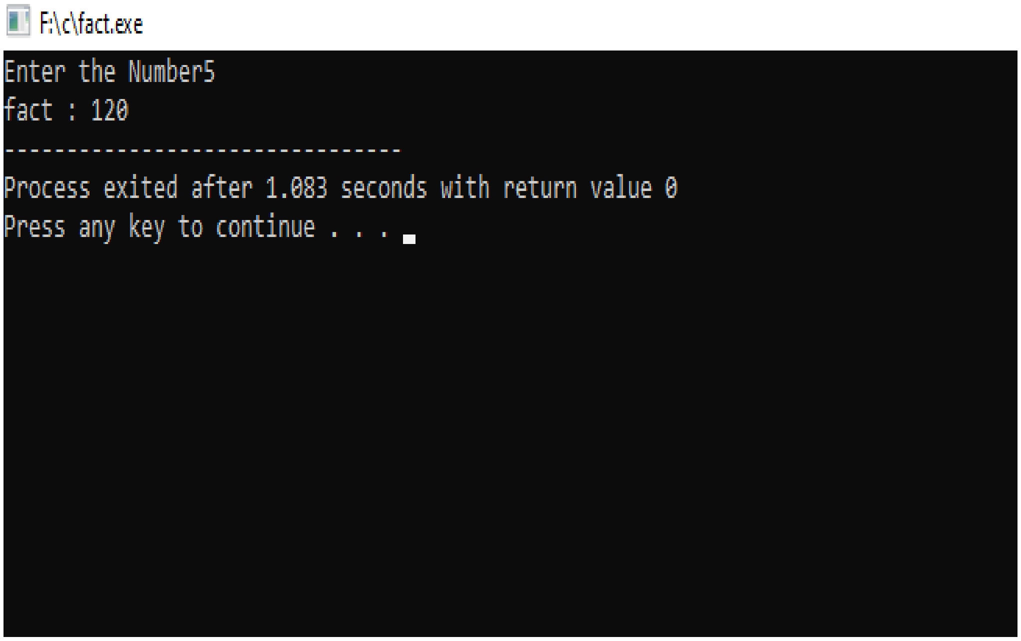
CSD CODING LAB

NAME : P . SAI TARRUN
REGD : 121910302010
SEC : B2

1) **Write a program for factorial of a number using recursion**

```
#include<stdio.h>
int fact(int n)
{
        if( n == 0|| n == 1)
        {
        return 1;
        }

        else
        {
                return (n * fact(n-1));
        }
}

main()
{
        int x;
        printf("Enter the Number");
        scanf("%d",&x);
        printf("fact : %d",fact(x));
}
```

OUTPUT

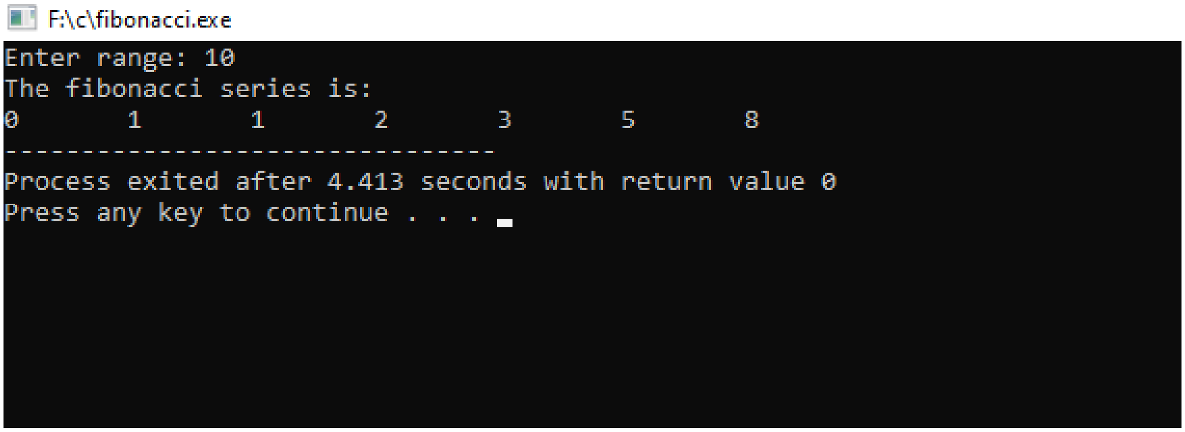


```
Enter the Number5
fact : 120
--------------------------------
Process exited after 1.083 seconds with return value 0
Press any key to continue . . .
```

2) Write a program for Fibonacci series using functions

```
#include<stdio.h>
void fibonacciSeries(int number)
{
  int a=0, b=1, c;
  while (a<=number)
  {
    printf("%d\t", a);
    c = a+b;
    a = b;
    b = c;
  }
}
int main()
{
  int number;
  printf("Enter range: ");
  scanf("%d", &number);
  printf("The fibonacci series is: \n");
  fibonacciSeries(number);
  return 0;
}
```

OUTPUT :



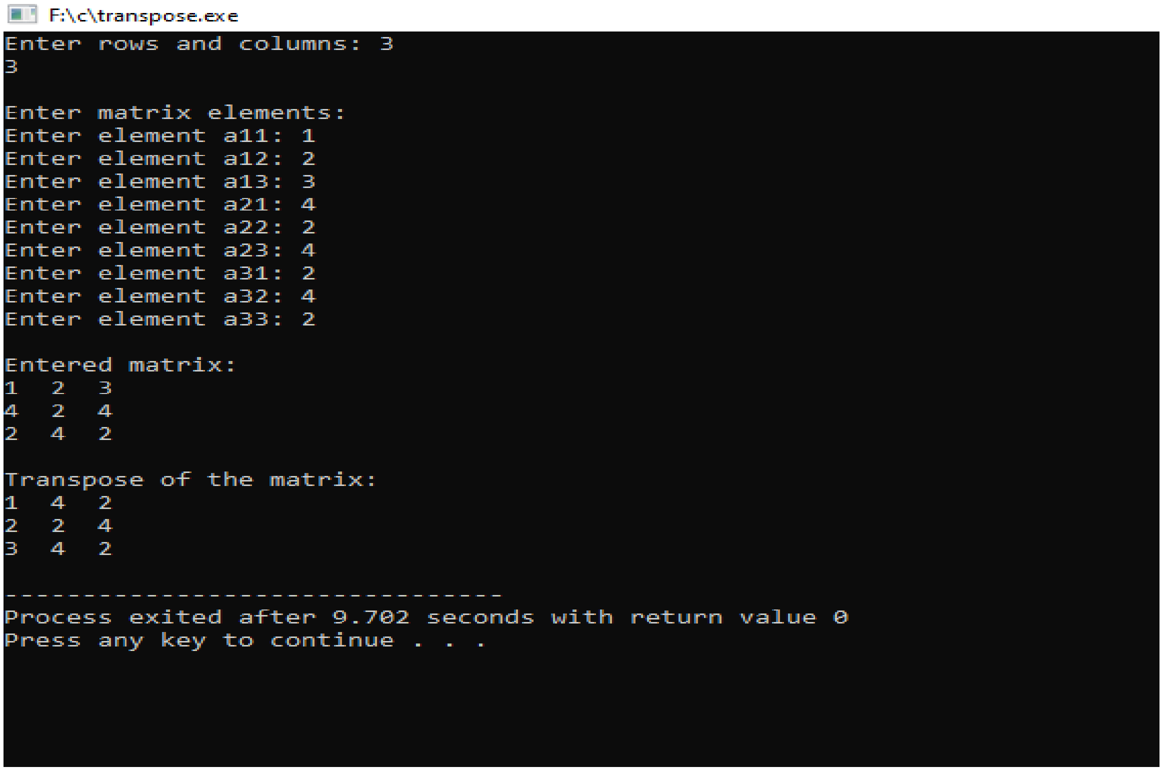
F:\c\fibonacci.exe

```
Enter range: 10
The fibonacci series is:
0       1       1       2       3       5       8
--------------------------------
Process exited after 4.413 seconds with return value 0
Press any key to continue . . .
```

3) Write a program for transpose of a matrix

```
#include <stdio.h>
int main() {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);
  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }
  printf("\nEntered matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }
  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

OUTPUT :

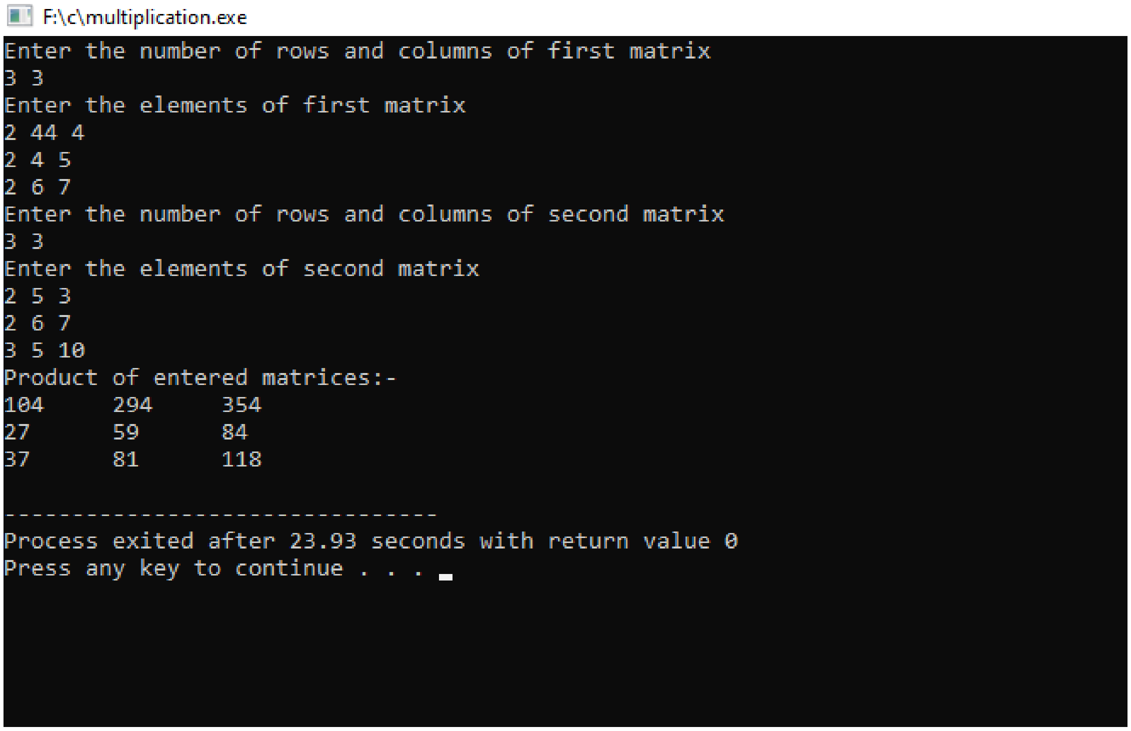


```
Enter rows and columns: 3
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 2
Enter element a13: 3
Enter element a21: 4
Enter element a22: 2
Enter element a23: 4
Enter element a31: 2
Enter element a32: 4
Enter element a33: 2

Entered matrix:
1   2   3
4   2   4
2   4   2

Transpose of the matrix:
1   4   2
2   2   4
3   4   2

--------------------------------
Process exited after 9.702 seconds with return value 0
Press any key to continue . . .
```

4) Write a program for matrix multiplication

```
#include <stdio.h>
int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter the number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter the elements of first matrix\n");
  for (  c = 0 ; c < m ; c++ )
    for ( d = 0 ; d < n ; d++ )
      scanf("%d", &first[c][d]);
  printf("Enter the number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);
  if ( n != p )
    printf("Matrices with entered orders can't be multiplied with each other.\n");
  else
  {
    printf("Enter the elements of second matrix\n");
    for ( c = 0 ; c < p ; c++ )
      for ( d = 0 ; d < q ; d++ )
        scanf("%d", &second[c][d]);
    for ( c = 0 ; c < m ; c++ )
  {
   for ( d = 0 ; d < q ; d++ )
   {
    for ( k = 0 ; k < p ; k++ )
    {
     sum = sum + first[c][k]*second[k][d];
    }
    multiply[c][d] = sum;
    sum = 0;
   }
  }
  printf("Product of entered matrices:-\n");
  for ( c = 0 ; c < m ; c++ )
  {
   for ( d = 0 ; d < q ; d++ )
    printf("%d\t", multiply[c][d]);
   printf("\n");
  }
 }
 return 0;
}
```

OUTPUT:



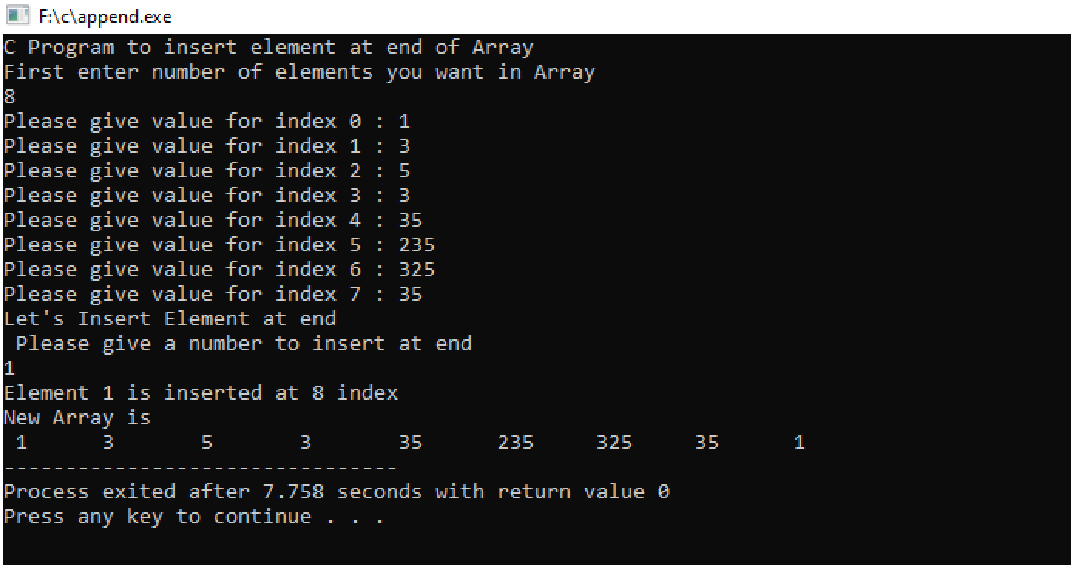
```
Enter the number of rows and columns of first matrix
3 3
Enter the elements of first matrix
2 44 4
2 4 5
2 6 7
Enter the number of rows and columns of second matrix
3 3
Enter the elements of second matrix
2 5 3
2 6 7
3 5 10
Product of entered matrices:-
104     294     354
27      59      84
37      81      118

--------------------------------
Process exited after 23.93 seconds with return value 0
Press any key to continue . . .
```

5) Write a program to append a new value to the end of an array

```
#include <stdio.h>
int main()
{
   int position, i, n, value,ch, arr[100];
   printf("C Program to insert element at end of Array\n");
   printf("First enter number of elements you want in Array\n");
   scanf("%d", &n);
   arr[n];
  for(i = 0; i < n; i++)
   {
      printf("Please give value for index %d : ",i);
      scanf("%d",&arr[i]);
```

```
    }
    printf("Let's Insert Element at end \n ");
    printf("Please give a number to insert at end \n");
    scanf("%d", &value);
    arr[n] = value;
    printf("Element %d is inserted at %d index \n",value,n);
    printf("New Array is \n ");

    for(i = 0; i < n+1; i++)
    {
      printf("%d \t",arr[i]);
    }
}
```

OUTPUT :

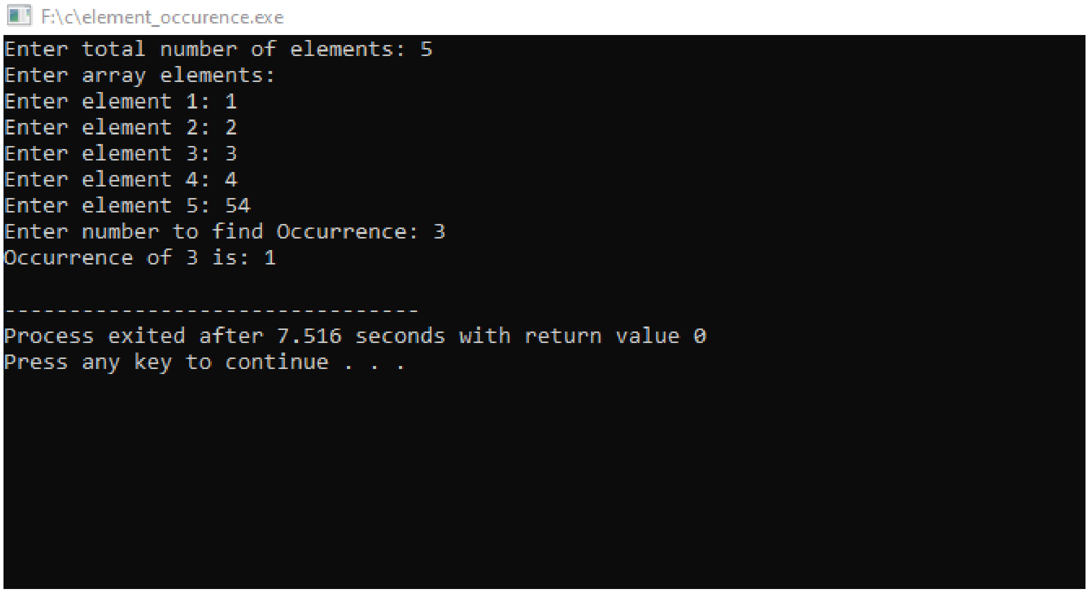


```
C Program to insert element at end of Array
First enter number of elements you want in Array
8
Please give value for index 0 : 1
Please give value for index 1 : 3
Please give value for index 2 : 5
Please give value for index 3 : 3
Please give value for index 4 : 35
Please give value for index 5 : 235
Please give value for index 6 : 325
Please give value for index 7 : 35
Let's Insert Element at end
 Please give a number to insert at end
1
Element 1 is inserted at 8 index
New Array is
 1      3       5       3       35      235     325     35      1
--------------------------------
Process exited after 7.758 seconds with return value 0
Press any key to continue . . .
```

6) Write a program to get the number of occurrences of a specified element in an array

```
#include <stdio.h>
#define MAX 100
int main()
{
    int arr[MAX], n, i;
   int num, count;
    printf("Enter total number of elements: ");
    scanf("%d", &n);
    //read array elements
    printf("Enter array elements:\n");
    for (i = 0; i < n; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
    printf("Enter number to find Occurrence: ");
    scanf("%d", &num);
    //count occurance of num
    count = 0;
    for (i = 0; i < n; i++) {
        if (arr[i] == num)
            count++;
    }
    printf("Occurrence of %d is: %d\n", num, count);
    return 0;
}
```

OUTPUT :

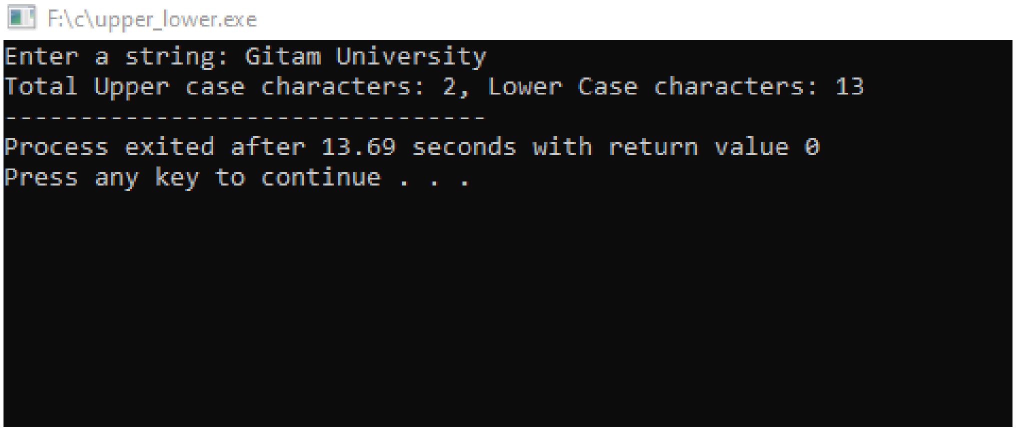


```
Enter total number of elements: 5
Enter array elements:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 54
Enter number to find Occurrence: 3
Occurrence of 3 is: 1

--------------------------------
Process exited after 7.516 seconds with return value 0
Press any key to continue . . .
```

7) **Write a program that accepts string and calculate the number of upper and lower case letters using functions**

```
#include <stdio.h>
int main()
{
   char str[100];
   int countL, countU;
   int counter;
   countL = countU = 0;
   printf("Enter a string: ");
   gets(str);

   for (counter = 0; str[counter] != NULL; counter++) {
       if (str[counter] >= 'A' && str[counter] <= 'Z')
           countU++;
       else if (str[counter] >= 'a' && str[counter] <= 'z')
           countL++;
   }
   printf("Total Upper case characters: %d, Lower Case characters: %d", countU,
countL);
   return 0;
}
```

OUTPUT :

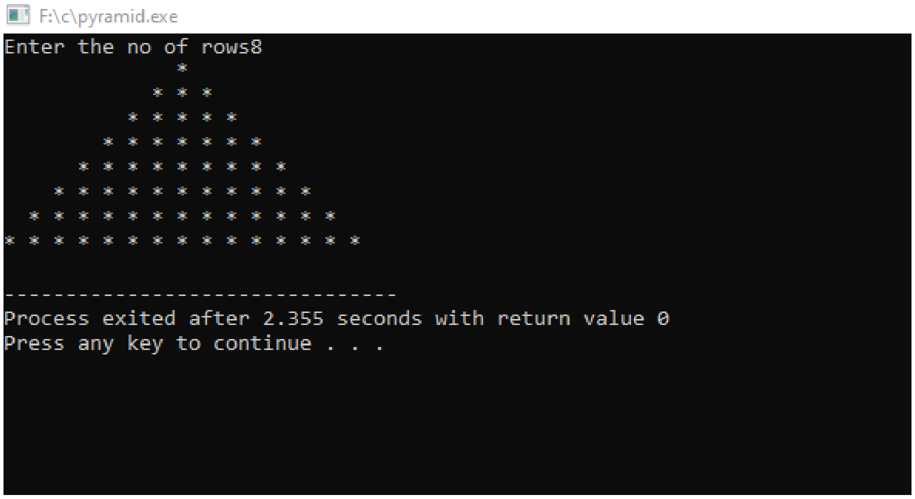


```
Enter a string: Gitam University
Total Upper case characters: 2, Lower Case characters: 13
--------------------------------
Process exited after 13.69 seconds with return value 0
Press any key to continue . . .
```

8) Program to print full pyramid using function

```
#include<stdio.h>
void pyramid()
{
    int i, j, k=0;
    int n;
    printf("Enter the no of rows");
    scanf("%d",&n);

    for(i = 1; i<= n; i++)
    {
        for(j = 1; j<= n-i; j++)
        {
            printf("  ");
        }
        for(k = 1; k <= (2* i-1); k++)
        {
            printf("* ");
        }
        printf("\n");
    }
}
main()
{
    pyramid();
}
```

OUTPUT :



```
Enter the no of rows8
              *
            * * *
          * * * * *
        * * * * * * *
      * * * * * * * * *
    * * * * * * * * * * *
  * * * * * * * * * * * * *
* * * * * * * * * * * * * * *


--------------------------------
Process exited after 2.355 seconds with return value 0
Press any key to continue . . .
```