



# Green University

## ASSIGNMENT SHEET Section : DE-221

Name : Khondokar Saim .....

ID : ..... 221902353 .....

Lecturer : Mahbubur Rahman .....

Subject : CSE-103 .....

Date : ..... 26 / 08 / 2022 .....

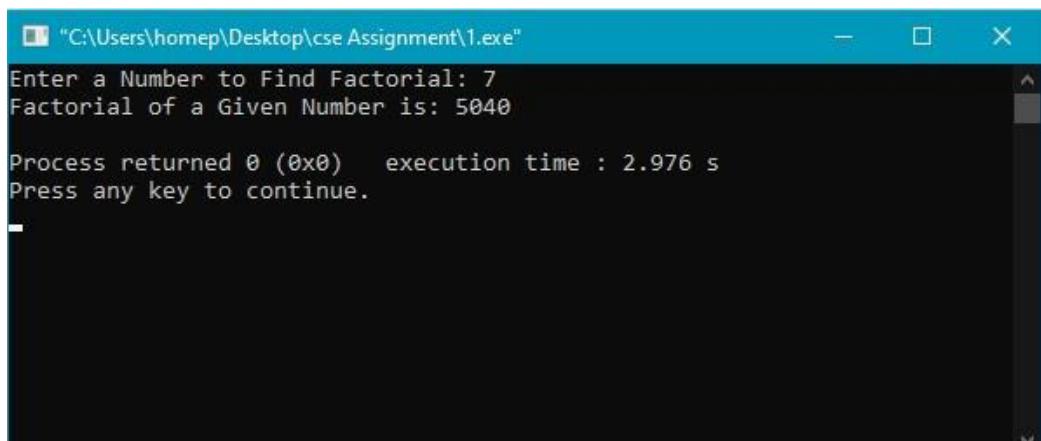
## Problem - 01 :

➤ Find factorial using a function.

Code :

```
/*
Find factorial using a function
*/
#include<stdio.h>
#include<math.h>
int main()
{
    printf("Enter a Number to Find Factorial: ");
    printf("Factorial of a Given Number is: %d \n",fact());
    return 0;
}
int fact()
{
    int i,fact=1,n;
    scanf("%d",&n);
    for(i=1; i<=n; i++)
    {
        fact=fact*i;
    }
    return fact;
}
```

Output :



## Problem - 02 :

➤ Reverse array using function.

Code :

```
/*Reverse array using function
*/
#include <stdio.h>

void reverse(int arr[10], int n)
{
    int i, tmp;
    for(i=0;i< n/2;i++)
    {
        tmp = arr[i];
        arr[i] = arr[n-1-i];
        arr[n-1-i] = tmp;
    }
}

int main()
{
    int arr[100], i, size;

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

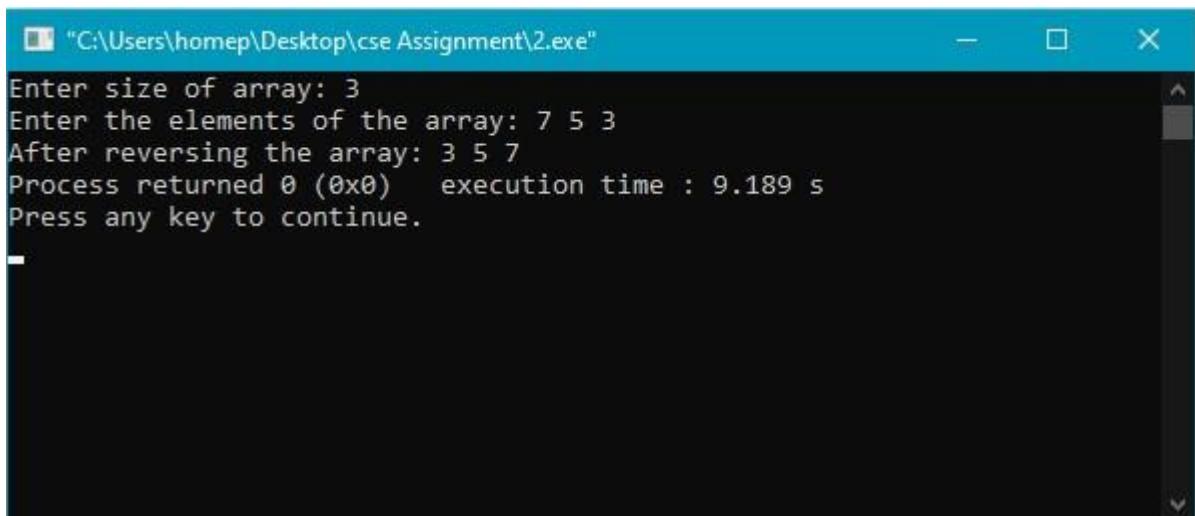
    printf("Enter the elements of the array: ");
    for (i = 0; i < size; i++)
        scanf("%d", &arr[i]);

    reverse(arr,size); //function call

    printf("After reversing the array: ");
    for(i=0;i < size;i++)
    {
```

```
{  
    printf("%d ", arr[i]);  
}  
return 0;  
}
```

## Output :



The screenshot shows a Windows command-line interface window titled "C:\Users\homep\Desktop\cse Assignment\2.exe". The window contains the following text output:

```
Enter size of array: 3  
Enter the elements of the array: 7 5 3  
After reversing the array: 3 5 7  
Process returned 0 (0x0)  execution time : 9.189 s  
Press any key to continue.
```

### Problem - 03 :

➤ Find maximum or minimum value of an array.

Code :

```
/*Find maximum or minimum value of an array
```

```
*/
```

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

```
#define MAX_SIZE 100
```

```
int main()
```

```
{
```

```
    int arr[MAX_SIZE];
```

```
    int i, max, min, size;
```

```
/* Input size of the array */
```

```
printf("Enter size of the array: ");
```

```
scanf("%d", &size);
```

```
/* Input array elements */
```

```
printf("Enter elements in the array: ");
```

```
for(i=0; i<size; i++)
```

```
{
```

```
    scanf("%d", &arr[i]);
```

```
}
```

```
/* Assume first element as maximum and minimum */
```

```
max = arr[0];
```

```
min = arr[0];
```

```
/*
```

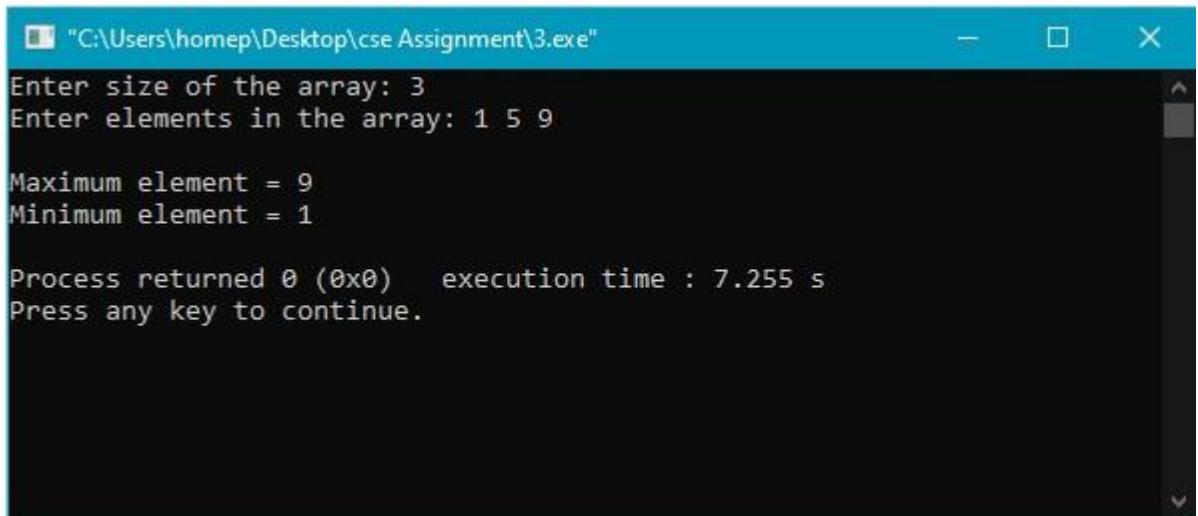
```
* Find maximum and minimum in all array elements.
```

```
*/
```

```
for(i=1; i<size; i++)
```

```
{  
    //If current element is greater than max  
    if(arr[i] > max)  
    {  
        max = arr[i];  
    }  
  
    //If current element is smaller than min  
    if(arr[i] < min)  
    {  
        min = arr[i];  
    }  
}  
  
//Print maximum and minimum element  
printf("\nMaximum element = %d\n", max);  
printf("Minimum element = %d\n", min);  
  
return 0;  
}
```

## Output :



The screenshot shows a terminal window titled "C:\Users\homep\Desktop\cse Assignment\3.exe". The window displays the following text:  
Enter size of the array: 3  
Enter elements in the array: 1 5 9  
Maximum element = 9  
Minimum element = 1  
Process returned 0 (0x0) execution time : 7.255 s  
Press any key to continue.

## Problem – 04 :

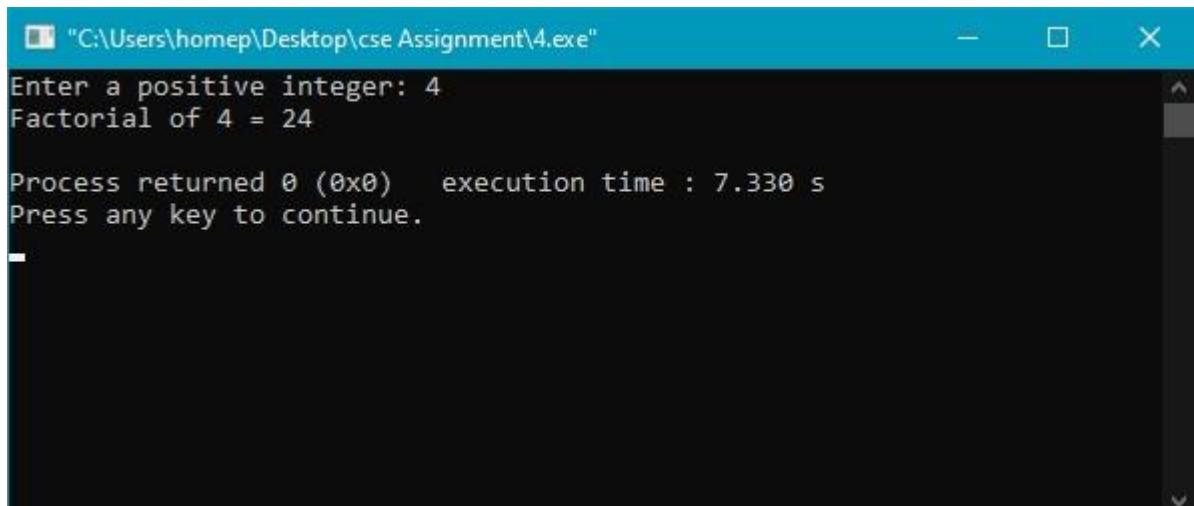
### ➤ Find Factorial of a Number Using Recursion.

Code :

```
/*Find Factorial of a Number Using Recursion
*/
#include<stdio.h>
long int multiplyNumbers(int n);
int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld\n", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n)
{
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

Output :



The screenshot shows a terminal window titled "C:\Users\homep\Desktop\cse Assignment\4.exe". The window displays the following text:

```
Enter a positive integer: 4
Factorial of 4 = 24

Process returned 0 (0x0)   execution time : 7.330 s
Press any key to continue.
```

## Problem – 05 :

➤ ASCII value of a character using Function.

Code :

```
/*
ASCII value of a character using Function
*/
```

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

```
#define TRUE 1
```

```
void clear_buffer( void )
```

```
{
```

```
    int ch;
```

```
    while( ( ch = getchar() ) != '\n' && ch != EOF );
```

```
}
```

```
int main()
```

```
{
```

```
    unsigned int ch;
```

```
    while( TRUE )
```

```
{
```

```
        printf("\nsingle char please - " );
```

```
        ch = getchar();
```

```
        clear_buffer();
```

```
        printf("%c - %d", ch, ch );
```

```
}
```

```
}
```

## Output :

```
single char please - d
d - 100
single char please - A
A - 65
single char please - i
i - 105
single char please - a
a - 97
single char please - t
t - 116
single char please - -
```

## Problem – 06 :

- Check an alphabet is vowel or consonant using Function.

### Code :

```
/*
Check an alphabet is vowel or consonant using Function
*/
#include <stdio.h>
int check_vowel(char a);
int main()
{
    char ch;
    printf("Please Enter an alphabet: ");
    scanf(" %c", &ch);

    if(check_vowel(ch)) {
        printf("\n%c is a VOWEL.\n", ch);
    }
    else {
        printf("\n%c is a CONSONANT.\n", ch);
    }
    return 0;
}

int check_vowel(char c)
{
    if (c >= 'A' && c <= 'Z')
        c = c + 32;

    if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
        return 1;

    return 0;
}
```

## Output :

### Part - 1

```
□ "C:\Users\homep\Desktop\cse Assignment\6.exe"
Please Enter an alphabet: U
U is a VOWEL.

Process returned 0 (0x0)    execution time : 1.401 s
Press any key to continue.
```

### Part - 2

```
□ "C:\Users\homep\Desktop\cse Assignment\6.exe"
Please Enter an alphabet: W
W is a CONSONANT.

Process returned 0 (0x0)    execution time : 6.802 s
Press any key to continue.
```

## Problem – 07 :

- Convert upper case to lower case and lower case to upper case using function

Code :

```
/*
Convert upper case to lower case and lower case to upper case using
function (UNDONE)
*/
#include <stdio.h>
#include <ctype.h>

void main()
{
    char sentence[100];
    int count, ch, i;

    printf("Enter a sentence \n");
    for (i = 0;(sentence[i] = getchar()) != '\n'; i++)
    {
        ;
    }
    sentence[i] = '\0';
    count = i;
    printf("\nThe given sentence is : %s", sentence);
    printf("\nCase changed sentence is: ");

    for (i = 0; i < count; i++)
    {
        ch = islower(sentence[i])? toupper(sentence[i]) :
        tolower(sentence[i]);
        putchar(ch);
    }
    printf("\n-----\n");
}
```

## Output :

```
C:\Users\homep\Desktop\cse Assignment\7.exe
Enter a sentence
kHONDOKAR sAIM

The given sentence is : khONDOKAR sAIM
Case changed sentence is: Khondokar Saim
-----
Process returned 0 (0x0) execution time : 11.987 s
Press any key to continue.
```

## Problem – 08 :

- Find out exponential without power method using Function.

Code :

```
/*
 Find out exponential without power method using Function
*/
#include <stdio.h>

int Pow(int X, int Y) {

    int power = 1, i;

    for (i = 1; i <= Y; ++i) {
        power = power * X;
    }

    return power;
}

int main() {

    long long int base, exponent;

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

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

    printf("%d ^ %d = %d", base, exponent, Pow(base, exponent));

}
```

## Output :

```
C:\Users\homep\Desktop\cse Assignment\8.exe"
Enter Base: 8
Enter Power: 2
8 ^ 2 = 64
Process returned 0 (0x0) execution time : 6.560 s
Press any key to continue.
```

## Problem – 09 :

- Do addition and subtraction operation of matrices using Function.

Code :

```
/*Do addition and subtraction operation of matrices using Function
*/
#include <stdio.h>
#define MAXROW    10
#define MAXCOL    10

/*User Define Function to Read Matrix*/
void readMatrix(int m[][MAXCOL],int row,int col)
{
    int i,j;
    for(i=0;i< row;i++)
    {
        for(j=0;j< col;j++)
        {
            printf("Enter element [%d,%d] : ",i+1,j+1);
            scanf("%d",&m[i][j]);
        }
    }
}

/*User Define Function to Read Matrix*/
void printMatrix(int m[][MAXCOL],int row,int col)
{
    int i,j;
    for(i=0;i< row;i++)
    {
        for(j=0;j< col;j++)
        {
            printf("%d\t",m[i][j]);
        }
    }
}
```

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

int main()
{
    int
a[MAXROW][MAXCOL],b[MAXROW][MAXCOL],result[MAXROW][MAX
COL];
    int i,j,r1,c1,r2,c2;

    printf("Enter number of Rows of matrix a: ");
    scanf("%d",&r1);
    printf("Enter number of Cols of matrix a: ");
    scanf("%d",&c1);

    printf("\nEnter elements of matrix a: \n");
    readMatrix(a,r1,c1);

    printf("Enter number of Rows of matrix b: ");
    scanf("%d",&r2);
    printf("Enter number of Cols of matrix b: ");
    scanf("%d",&c2);

    printf("\nEnter elements of matrix b: \n");
    readMatrix(b,r2,c2);

/*sum and sub of Matrices*/
if(r1==r2 && c1==c2)
{
    for(i=0;i< r1;i++)
    {
        for(j=0;j< c1;j++)
        {
```

```

{
    result[i][j]=a[i][j]+b[i][j];
}
}

/*print matrix*/
printf("\nMatrix after adding (result matrix):\n");
printMatrix(result,r1,c1);

/*Subtracting two matrices a and b, and result storing in matrix
result*/
for(i=0;i< r1;i++)
{
    for(j=0;j< c1;j++)
    {
        result[i][j]=a[i][j]-b[i][j];
    }
}

/*print matrix*/
printf("\nMatrix after subtracting (result matrix):\n");
printMatrix(result,r1,c1);

}

else
{
    printf("\nMatrix can not be added, Number of Rows & Cols are
Different");
}
return 0;
}

```

## Output :

```
C:\Users\homep\Desktop\cse Assignment\9.exe"
```

```
Enter element [2,2] : 55
Enter element [2,3] : 66
Enter element [3,1] : 77
Enter element [3,2] : 88
Enter element [3,3] : 99
Enter number of Rows of matrix b: 3
Enter number of Cols of matrix b: 3
```

```
Enter elements of matrix b:
Enter element [1,1] : 1
Enter element [1,2] : 2
Enter element [1,3] : 3
Enter element [2,1] : 4
Enter element [2,2] : 5
Enter element [2,3] : 6
Enter element [3,1] : 7
Enter element [3,2] : 8
Enter element [3,3] : 9
```

```
Matrix after adding (result matrix):
```

```
12      24      36
48      60      72
84      96      108
```

```
Matrix after subtracting (result matrix):
```

```
10      20      30
40      50      60
70      80      90
```

```
Process returned 0 (0x0)  execution time : 64.812 s
```

```
Press any key to continue.
```

## Problem – 10 :

➤ Do multiplication operation of matrices using Function.

Code :

```
/*Do multiplication operation of matrices using Function
*/
#include<stdio.h>
void multiply(int mat1[12][12],int mat2[12][12],int ,int ,int );
void main()
{
    int mat1[12][12],mat2[12][12];
    int i,j,k,m,n,p;
    printf("Enter the number of rows and columns for 1st matrix\n");
    scanf("%d%d",&m,&n);
    printf("Enter the elements of the 1st matrix\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&mat1[i][j]);
        }
    }

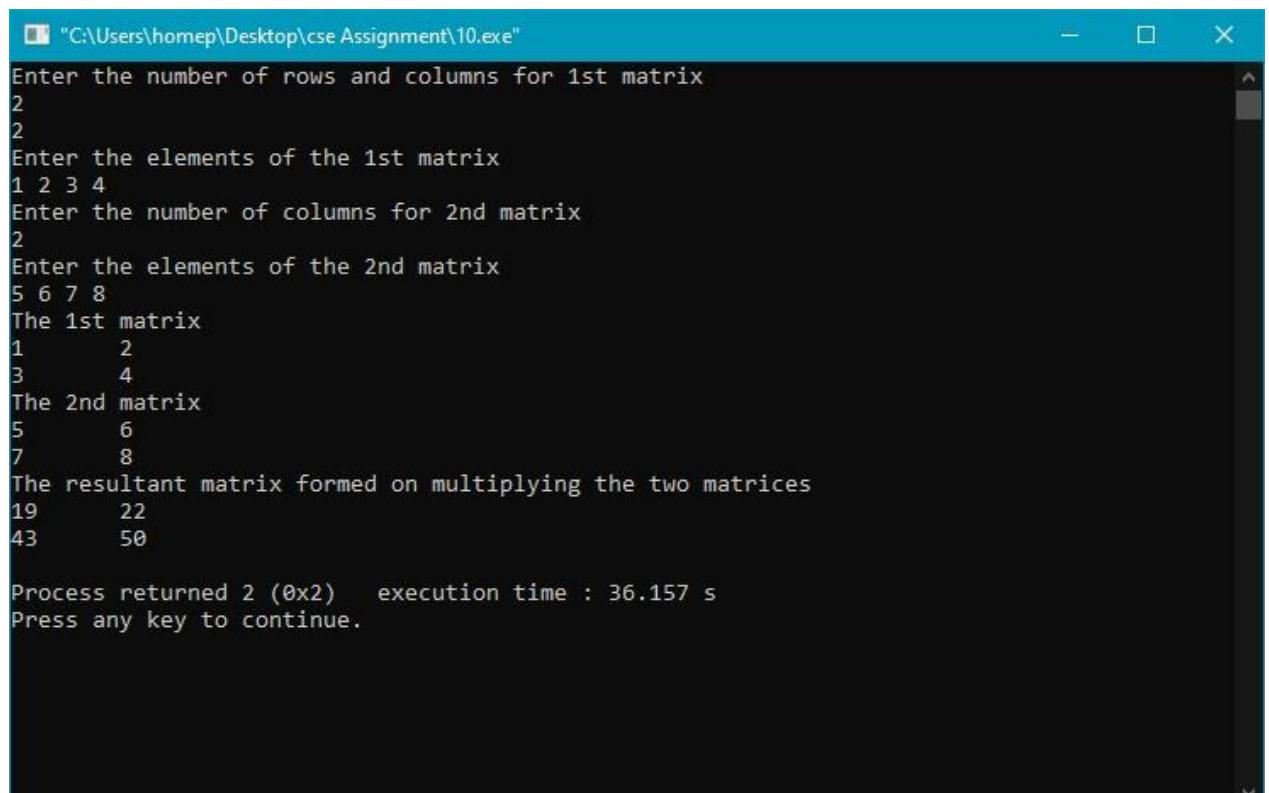
    printf("Enter the number of columns for 2nd matrix\n");
    scanf("%d",&p);
    printf("Enter the elements of the 2nd matrix\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<p;j++)
        {
            scanf("%d",&mat2[i][j]);
        }
    }
}
```

```
printf("The 1st matrix\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%d\t",mat1[i][j]);
    }
    printf("\n");
}
printf("The 2nd matrix\n");
for(i=0;i<n;i++)
{
    for(j=0;j<p;j++)
    {
        printf("%d\t",mat2[i][j]);
    }
    printf("\n");
}
multiply(mat1,mat2,m,n,p);
}

void multiply(int mat1[12][12],int mat2[12][12],int m,int n,int p)
{
    int mul[12][12],i,j,k;
    for(i=0;i<m;i++)
    {
        for(j=0;j<p;j++)
        {
            mul[i][j]=0;
            for(k=0;k<n;k++)
            {
                mul[i][j]=mul[i][j]+mat1[i][k]*mat2[k][j];
            }
        }
    }
}
```

```
printf("The resultant matrix formed on multiplying the two  
matrices\n");  
for(i=0;i<m;i++)  
{  
    for(j=0;j<p;j++)  
    {  
        printf("%d\t",mul[i][j]);  
    }  
    printf("\n");  
}  
}
```

## Output :



The screenshot shows a terminal window titled "C:\Users\homep\Desktop\cse Assignment\10.exe". The window displays the following interaction:

```
Enter the number of rows and columns for 1st matrix
2
2
Enter the elements of the 1st matrix
1 2 3 4
Enter the number of columns for 2nd matrix
2
Enter the elements of the 2nd matrix
5 6 7 8
The 1st matrix
1      2
3      4
The 2nd matrix
5      6
7      8
The resultant matrix formed on multiplying the two matrices
19      22
43      50

Process returned 2 (0x2)   execution time : 36.157 s
Press any key to continue.
```