

# LAB ASSIGNMENT - 4

```
*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/

/*Q1. Write suitable function and corresponding program to test them for the following:
a. Compute  $X^n$ , where X is any valid number and n is an integer value.
b. Swap values of two integer variables with (a) using a temporary variable, (b) without using a temporary variable.
c. Compute the GCD of two integers and return the result to the calling function.
d. Compute and returns the sum of n elements of an integer array.
*/
#include<stdio.h>
float power( float x , int n )
{
    if( n == 1 )
        return x;
    return x * power( x , n - 1 );
}
void swap(int *a , int *b )
{
    int t = *a ;
    *a = *b ;
    *b = t ;
}
void swaps(int *a , int *b )
{
    *a = *a + *b ;
    *b = *a - *b;
    *a = *a - *b;
}
int gcd( int a , int b )
{
    if(a % b == 0)
        return b;
    return gcd( b , a % b );
}
int sum( int a[] , int n )
{
    int s = 0 ;
    for( int i = 0 ; i < n ; i++ )
    {
        s += a[i];
    }
}
```

```

    return s ;
}
int main()
{
    char ch;
    printf("Enter your choice 'a', 'b' , 'c' or 'd' : ");
    scanf(" %c", &ch );
    float x ; int n ;
    int i;
    int a , b ;
    int n1 , n2 ;
    if(ch == 'd')
    {
        printf("Enter the size of integer array : ");
        scanf("%d", &n );
        int arr[n];
        printf("Enter the elements os the array :\n");
        for ( i = 0 ; i < n ; i++ )
        {
            scanf("%d", &arr[i]);
        }
        printf("Sum of the elements : %d\n", sum(arr,n));
    }
    switch(ch)
    {
        case 'a' : printf("Enter the value of base and index ");
                    scanf("%f%d", &x , &n );
                    printf("%f ^ %d = %f", x , n , power(x,n) );
                    break;
        case 'b' : printf("Enter 1 or 2 : ");
                    scanf("%d", &i );
                    printf("Enter two numbers : ");
                    scanf("%d%d", &a , &b );
                    if( i == 1 )
                        swap( &a , &b );
                    else if( i == 2 )
                        swaps( &a , &b );
                    else
                    {
                        printf("~Invalid Input~");
                        break;
                    }
                    printf("The swapped values are : %d and %d\n", a , b );
                    break;
        case 'c' : printf("Enter two numbers : ");
                    scanf("%d %d", &n1, &n2 );
                    printf("The Greatest Common Divisor of %d and %d id %d.\n",
n1 , n2 , gcd(n1,n2) );

```

```

        break;
    case 'd' : break;
    default : printf("! Invalid Input !");
              break;
}
return 0;
}

```

OUTPUT:

Enter your choice 'a', 'b', 'c' or 'd' : a

Enter the value of base and index 5.2 3

5.200000 ^ 3 = 140.607985

Enter your choice 'a', 'b', 'c' or 'd' : b

Enter 1 or 2 : 2

Enter two numbers : 15 63

The swapped values are : 63 and 15

Enter your choice 'a', 'b', 'c' or 'd' : c

Enter two numbers : 12 64

The Greatest Common Divisor of 12 and 64 is 4.

Enter your choice 'a', 'b', 'c' or 'd' : d

Enter the size of integer array : 5

Enter the elements of the array :

1 9 7 3 2

Sum of the elements : 22

Enter your choice 'a', 'b', 'c' or 'd' : o

! Invalid Input !

```

/*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/

/*Q2. Write a C recursive function to find the factorial of an integer N.*/

#include<stdio.h>

```

```

int factorial(int n )
{
    if(n == 1)
    {
        return 1;
    }
    return n*factorial(n-1);
}
int main()
{
    int n;
    printf("Enter a number : ");
    scanf("%d",&n );
    printf("%d! = %d", n , factorial(n));
    return 0;
}

```

OUTPUT :

Enter a number : 7

7! = 5040

```

/*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/

/*Q3. Write C functions for the following problems:
a. For a natural number find out its factors.
b. For a range of numbers, say, 1 to N, find out the factors of each number and
determine the one that has got a maximum number of factors.*/

#include<stdio.h>

int num_of_fac(int n, int i)
{
    if(i==n)
        return 1;
    if(n%i == 0)
        return 1 + num_of_fac( n , i + 1 );
    else
        return num_of_fac( n , i + 1);
}

void fac(int n , int i)
{
    if(i==n+1)
        return;
    if( n % i == 0 )
        printf("%d\t",i);
    fac( n , i+1 );
}

```

```

}
int max_fac(int n , int max , int x)
{
    if(n == 0)
        return x;
    if( max < num_of_fac( n , 1 ) )
    {
        max = num_of_fac( n , 1 );
        x = n;
    }
    max_fac( n - 1 , max , x );
}
int main()
{
    printf("Enter a natural number : ");
    int n ;
    scanf("%d",&n);
    printf("The factors of %d are :\t", n);
    fac( n , 1 );
    printf("\n");
    for(int i = 1 ; i <= n ; i++)
    {
        printf("Factors of %d :\t", i);
        fac(i,1);
        printf("\n");
    }
    printf("Number with maximum number of factors : %d", max_fac(n , 0 , 0));
    return 0;
}

```

OUTPUT :

Enter a natural number : 45

The factors of 45 are : 1    3    5    9    15    45

Factors of 1 : 1

Factors of 2 : 1    2

Factors of 3 : 1    3

Factors of 4 : 1    2    4

Factors of 5 : 1    5

Factors of 6 : 1    2    3    6

Factors of 7 : 1    7

Factors of 8 : 1    2    4    8

Factors of 9 : 1    3    9

Factors of 10 : 1	2	5	10					
Factors of 11 : 1	11							
Factors of 12 : 1	2	3	4	6	12			
Factors of 13 : 1	13							
Factors of 14 : 1	2	7	14					
Factors of 15 : 1	3	5	15					
Factors of 16 : 1	2	4	8	16				
Factors of 17 : 1	17							
Factors of 18 : 1	2	3	6	9	18			
Factors of 19 : 1	19							
Factors of 20 : 1	2	4	5	10	20			
Factors of 21 : 1	3	7	21					
Factors of 22 : 1	2	11	22					
Factors of 23 : 1	23							
Factors of 24 : 1	2	3	4	6	8	12	24	
Factors of 25 : 1	5	25						
Factors of 26 : 1	2	13	26					
Factors of 27 : 1	3	9	27					
Factors of 28 : 1	2	4	7	14	28			
Factors of 29 : 1	29							
Factors of 30 : 1	2	3	5	6	10	15	30	
Factors of 31 : 1	31							
Factors of 32 : 1	2	4	8	16	32			
Factors of 33 : 1	3	11	33					
Factors of 34 : 1	2	17	34					
Factors of 35 : 1	5	7	35					
Factors of 36 : 1	2	3	4	6	9	12	18	36
Factors of 37 : 1	37							
Factors of 38 : 1	2	19	38					
Factors of 39 : 1	3	13	39					
Factors of 40 : 1	2	4	5	8	10	20	40	

Factors of 41 : 1    41

Factors of 42 : 1    2    3    6    7    14    21    42

Factors of 43 : 1    43

Factors of 44 : 1    2    4    11    22    44

Factors of 45 : 1    3    5    9    15    45

Number with maximum number of factors : 36

```
/*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/
```

```
/*Q4. Write a C function reverse (s) to reverse the string s, where s is an argument in the function  
reverse (s).*/
```

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

```
#include<stdlib.h>
```

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

```
void reverse(char s[])
```

```
{
```

```
    char newstr[strlen(s)];
```

```
    for( int i = 0 , j = strlen(s)-1 ; i <= strlen(s)-1 ; i++ , j-- )
```

```
    {
```

```
        newstr[j]=s[i];
```

```
    }
```

```
    newstr[strlen(s)-1] ='\0';
```

```
    strcpy( s, newstr );
```

```
    return ;
```

```
}
```

```
int main()
```

```
{
```

```
    int size;
```

```
    printf("Enter the size of the string and the string : ");
```

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

```
    char s[size+1];
```

```
    gets(s);
```

```
    reverse(s);
```

```
    puts(s);
```

```
    return 0;
```

```
}
```

OUTPUT :

Enter the size of the string and the string : 18 EGELLOC\_EHT\_NEPOER

REOPEN\_THE\_COLLEGE

```

/*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/

/*Q5. Write a C function that takes input a two-
dimensional array of integers and find the largest
integer among them and return it to calling function.*/

#include<stdio.h>

int main()
{
    int r , c ;
    printf("Enter the number of rows and columns respectively : ");
    scanf("%d%d", &r , &c );
    int arr[r][c];
    printf("Enter the elements of the array : \n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            scanf("%d" , &arr[i][j]);
        }
    }
    printf("The array : \n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            printf("%d\t" , arr[i][j]);
        }
        printf("\n");
    }
    int max = arr[0][0];
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            if(arr[i][j] > max)
                max = arr[i][j] ;
        }
    }
    printf("The largest number in the array : %d ", max );
    return 0;
}

```

OUTPUT :

Enter the number of rows and columns respectively : 4 5

Enter the elements of the array :



1 3 8 4 9

4 8 6 2 7

5 1 2 8 9

7 4 2 3 6

The array :

1    3    8    4    9

4    8    6    2    7

5    1    2    8    9

7    4    2    3    6

The largest number in the array : 9

```
/*Lab Assignment-3 10-03-2021 Tathagata Ghosh*/

/*Q6. Write a C function to multiple two two-
dimensional matrices A and B and store the result
in another matrix C.*/
#include<stdio.h>

int main()
{
    printf("Enter the number of rows and columns of 1st matrix respectively : \n");
    int n1,n2,n3;
    scanf("%d%d", &n1 , &n2 );
    printf("Enter the number of columns of 2nd matrix : \n");
    scanf("%d", &n3);
    printf("Enter the values in the 1st matrix : \n");
    int a[n1][n2];
    int b[n2][n3];
    for(int i=0;i<n1;i++)
    {
        for(int j=0;j<n2;j++)
        {
            scanf("%d" , &a[i][j]);
        }
    }
    printf("Enter the values in the 2nd matrix : \n") ;
    for(int i=0;i<n2;i++)
    {
        for(int j=0;j<n3;j++)
        {
            scanf("%d" , &b[i][j]);
        }
    }
}
```

```

printf("The 1st matrix : \n");
for(int i=0;i<n1;i++)
{
    for(int j=0;j<n2;j++)
    {
        printf("%d\t", a[i][j]);
    }
    printf("\n");
}
printf("The 2nd matrix : \n");
for(int i=0;i<n2;i++)
{
    for(int j=0;j<n3;j++)
    {
        printf("%d\t" , b[i][j]);
    }
    printf("\n");
}
int c[n1][n3];
for(int i=0;i<n1;i++)
{
    for(int j=0;j<n3;j++)
    {
        c[i][j]=0;
    }
}
for(int i=0;i<n1;i++)
{
    for(int j=0;j<n3;j++)
    {
        for(int k=0;k<n2;k++)
        {
            c[i][j]+=(a[i][k]*b[k][j]);
        }
    }
}
printf("The resultant matrix after multiplication : \n");
for(int i=0;i<n1;i++)
{
    for(int j=0;j<n3;j++)
    {
        printf("%d\t" , c[i][j]);
    }
    printf("\n");
}
return 0;
}

```

OUTPUT :

Enter the number of rows and columns of 1st matrix respectively :

2 3

Enter the number of columns of 2nd matrix :

4

Enter the values in the 1st matrix :

1

9

3

6

5

4

Enter the values in the 2nd matrix :

7

2

6

9

4

1

8

5

3

2

10

84

The 1st matrix :

1    9    3

6    5    4

The 2nd matrix :

7    2    6    9

4    1    8    5

3    2    10    84

The resultant matrix after multiplication :

52    17    108    306

74    25    116    415

GitHub Repository : <https://github.com/Tathagata-Ghosh-Developer/Lab-Assignments>