

## CSB Lab-8

**Q1)** Declare a union containing 5 string variables (Name, House Name, City Name, State and Pin code) each with a length of C\_SIZE (user defined constant). Then, read and display a person using a variable of the union.

```
#include<stdio.h>
#include<string.h>
struct person{
    char name[30];
    char house_name[20],city_name[25],state[20],pin_code[10];
};
void main(){
    struct person add;
    printf("Enter name: ");
    scanf("%s", &add.name);
    printf("Enter house name: ");
    scanf("%s", &add.house_name);
    printf("Enter city name: ");
    scanf("%s", &add.city_name);
    printf("Enter state: ");
    scanf("%s", &add.state);
    printf("Enter pincode: ");
    scanf("%s", &add.pin_code);
    printf("Following information you have entered about the person:\n");
    printf("Name : %s", add.name);
    printf("\nHouse Name : %s", add.house_name);
    printf("\nCity Name : %s", add.city_name);
    printf("\nState : %s", add.state);
    printf("\nPin Code : %s", add.pin_code);
}
```

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS G:\Nitin\Code Blocks\Calm> gcc Address.c
PS G:\Nitin\Code Blocks\Calm> .\a.exe
Enter name: Steve
Enter house name: Ich
Enter city name: Paris
Enter state: Dice
Enter pincode: 47389
Following information you have entered about the person:
Name : Steve
House Name : Ich
City Name : Paris
State : Dice
Pin Code : 47389
```

**Q2)** Find the factorial of a given Natural Number n using recursive and non-recursive function.

```
#include<stdio.h>
int num;
int fac(int fac1, int fac0){
    if (fac0!=0){
        fac1 = fac1*fac0;
        fac(fac1, fac0-1);
    }
    else{return fac1;}
}
int main(){
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("The Factorial of %d is %d.", num, fac(num, num-1));
}
```

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

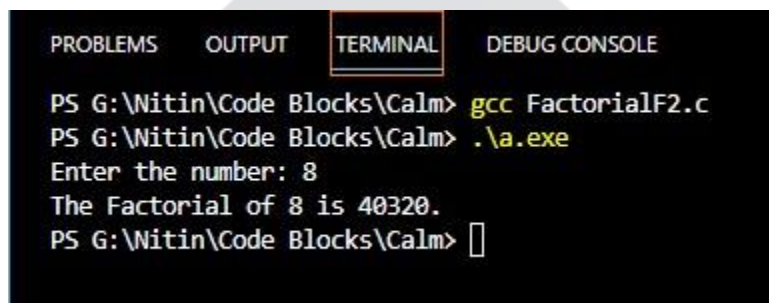
PS G:\Nitin\Code Blocks\Calm> gcc FactorialF1.c
PS G:\Nitin\Code Blocks\Calm> .\a.exe
Enter the number: 5
The Factorial of 5 is 120.
PS G:\Nitin\Code Blocks\Calm> 
```

```

#include<stdio.h>
int num, i = 1, res = 1;
int fac(int num)
{
    while (i<=num){
        res = res*i;
        i++;
    }
    return res;
}

int main(){
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("The Factorial of %d is %d.", num, fac(num));
}

```



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active, displaying the following commands and output:

```

PS G:\Nitin\Code Blocks\Calm> gcc FactorialF2.c
PS G:\Nitin\Code Blocks\Calm> .\a.exe
Enter the number: 8
The Factorial of 8 is 40320.
PS G:\Nitin\Code Blocks\Calm> 

```

**Q3)** Read a string (word), store it in an array and obtain its reverse by using a user defined function.

```

#include <stdio.h>
#include <string.h>

void revstr(char *str1)
{
    int i, len, temp;
    len = strlen(str1);
    for (i = 0; i < len/2; i++){
        temp = str1[i];
        str1[i] = str1[len - i - 1];
        str1[len - i - 1] = temp;
    }
}

```

```

int main(){
    char str[50];
    printf ("Enter the string: ");
    scanf("%s", &str);
    revstr(str);
    printf ("After reversing the string: %s", str);
}

```

PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
<pre> PS G:\Nitin\Code Blocks\Calm&gt; gcc ReverseS.c PS G:\Nitin\Code Blocks\Calm&gt; .\a.exe Enter the string: Mirror After reversing the string: rorriM PS G:\Nitin\Code Blocks\Calm&gt;  </pre>			

**Q4)** Write a menu driven program for performing matrix addition, multiplication and finding the transpose. Use functions to (i) read a matrix, (ii) find the sum of two matrices, (iii) find the product of two matrices, (iv) find the transpose of a matrix and (v) display a matrix.

```

#include<stdio.h>

#include<stdlib.h>
void add(int m[3][3], int n[3][3], int sum[3][3])
{
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
            sum[i][j] = m[i][j] + n[i][j];
}

void multiply(int m[3][3], int n[3][3], int result[3][3])
{
    for(int i=0; i < 3; i++){
        for(int j=0; j < 3; j++){
            result[i][j] = 0;
            for (int k = 0; k < 3; k++)
                result[i][j] += m[i][k] * n[k][j];
        }
    }
}

```

```

void transpose(int matrix[3][3], int trans[3][3])
{
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            trans[i][j] = matrix[j][i];
}

void display(int matrix[3][3])
{
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++)
            printf("%d\t",matrix[i][j]);
        printf("\n");
    }
}

int main(){
    int a[][3] = { {5,6,7}, {8,9,10}, {3,1,2} };
    int b[][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int c[3][3];
    printf("First Matrix:\n");
    display(a);
    printf("Second Matrix:\n");
    display(b);
    int choice;
    do{
        printf("\nChoose the matrix operation\n");
        printf("1. Addition\n");
        printf("2. Multiplication\n");
        printf("3. Transpose\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                add(a, b, c);
                printf("Sum of matrix: \n");
                display(c);
                break;
            case 2:
                multiply(a, b, c);
                printf("Multiplication of matrix: \n");
                display(c);
                break;
            case 3:
                printf("Transpose of the first matrix: \n");
                transpose(a, c);
                display(c);

```

```

        printf("Transpose of the second matrix: \n");
        transpose(b, c);
        display(c);
        break;
default:
    printf("Invalid input.\n");
    printf("Please enter the correct input.\n");
}
}while(1);
return 0;
}

```

PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE

```

PS G:\Nitin\Code Blocks\Calm> .\a.exe
First Matrix:
5      6      7
8      9     10
3      1      2
Second Matrix:
1      2      3
4      5      6
7      8      9

Enter your choice: 1
Sum of matrix:
6      8     10
12     14     16
10     9      11

Choose the matrix operation
1. Addition
2. Multiplication
3. Transpose
Enter your choice: 2
Multiplication of matrix:
78     96    114
114    141    168
21     27     33

Choose the matrix operation
1. Addition
2. Multiplication
3. Transpose
Enter your choice: 3
Transpose of the first matrix:
5      8      3
6      9      1
7     10      2
Transpose of the second matrix:
1      4      7
2      5      8
3      6      9

```