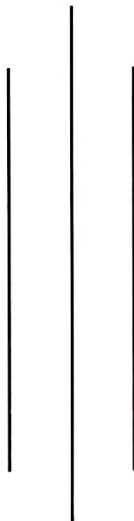




Tribhuvan University
Institute of Science and Technology
Patan multiple campus
Patandhoka, Lalitpur



Lab report
on
Discrete Structure (BIT152)

Submitted by
Dolindra Bahadur Raut
Roll no: 21
BIT-1st year/2nd semester

Date of Submission: 2081/...../.....

Submitted to
Shiv Kumar Mahato

.....

Signature

Tools used:

- **Operating system:** Microsoft windows
- **Programming language:** C++
- **Packages/IDE:** Dev C/C++

INDEX

S. N	Lab questions	Date of submission	Signature
1.	Write a program to find the length of string using strlen function.		
2.	Write a program to check a number is prime or not.		
3.	Write a program to find the sum of natural number using recursion.		
4.	Write a program to find the Fibonacci series using recursion.		
5.	Write a program to find the factorial of n number using recursion.		
6.	Write a program to find the sum of digits of any given positive number.		
7.	Write a program to find the greatest and smallest number in an array.		
8.	Write a program to find the GCD (Greatest common divisor) of the two positive integers using Euclidean algorithm.		
9.	Write a program to simulate the use of MOD operator.		
10.	Write a program to perform the following operations in zero-one matrix. i. Disjunction ii. Conjunction		
11.	Write a program to find the join and meet of Boolean matrix.		

1. Write a program to find the length of string using strlen function.



Program:

```
#include<iostream>
#include<cstring>
using namespace std;
int main () {
    char name[40];
    cout << "Enter any string : ";
    gets(name);
    cout << "Length of the string = " <<
        strlen(name);
    return 0;
}
```

Output:

```
Enter any string : DolihdeqUBahadyeURaut
Length of the string = 24
```

Note:- The sign "U" refers to the white space pressed by the user.

2. Write a program to check a number is prime or not.

⇒ program :-

```
#include<iostream>
using namespace std;
int main() {
    int num, flag = 1, i;
    cout << "Enter any positive number: ";
    cin >> num;
    if (num == 0 || num == 1) {
        flag = 0;
    }
    for (i = 2; i <= num/2; i++) {
        if (num % i == 0) {
            flag = 0;
            break;
        }
    }
    if (flag == 1) {
        cout << num << " is a prime number.";
    } else {
        cout << num << " is not a prime number.";
    }
    return 0;
}
```

Output :-

Enter any positive number: 5 5 is a prime number.
--

Enter any positive number: 8 8 is not a prime number.
--

3. Write a program to find the sum of natural numbers using recursion.

⇒ program :-

```
#include<iostream>
using namespace std;

int sum (int n){
    if (n <= 0){
        return 0;
    }
    else {
        return (n + sum(n-1));
    }
}

int main() {
    int num, result;
    cout << "Enter the value of n: ";
    cin >> num;
    result = sum(num);
    cout << "Sum of " << num << " natural numbers = " << result;
    return 0;
}
```

Output:-

```
Enter the value of n: 10
Sum of 10 natural numbers = 55
```

```
Enter the value of n: 5
Sum of 5 natural numbers = 15.
```

4. Write a program to find the Fibonacci series using recursion.

⇒ program :-

```
#include<iostream>
using namespace std;
int fibonacci (int num) {
    if ((num == 1) || (num == 0)) {
        return (num);
    }
    else {
        return (fibonacci (num - 1) + fibonacci (num - 2));
    }
}
int main() {
    int n, i;
    cout << "Enter the no. of terms for
    Fibonacci series: ";
    cin >> n;
    for (i = 0; i < n; i++) {
        cout << fibonacci (i) << " ";
    }
    return 0;
}
```

Output :-

```
Enter the no. of terms for Fibonacci series: 95
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

5. Write a program to find factorial of n number using recursion.

⇒ program :-

```
# include<iostream>
using namespace std;
int factorial (int n) {
    if (n <= 0) {
        return 1;
    }
    else {
        return (n * factorial (n-1));
    }
}
int main() {
    int num;
    cout << "Enter the number that you want to calculate factorial of : ";
    cin >> num;
    cout << "Factorial of " << num << " = " << factorial (num);
    return 0;
}
```

Output :-

Enter the number that you want to calculate factorial of : 5

Factorial of 5 = 120

6. Write a program to find the sum of digits of any given positive number.

⇒ Program:

```
#include<iostream>
using namespace std;

int main() {
    int num, sum = 0, r, original_num;
    cout << "Enter any positive number: ";
    cin >> num;
    original_num = num;
    while (num > 0) {
        r = num % 10;
        sum = sum + r;
        num = num / 10;
    }
    cout << "sum of digits of " << original_num <<
        " = " << sum;
    return 0;
}
```

Output:

```
Enter any positive number: 45
Sum of digits of 45 = 9
```

```
Enter any positive number: 523
Sum of digits of 523 = 10
```

7. Write a program to find the greatest and smallest number in an array.

⇒ program:

```
#include<iostream>
using namespace std;

int main() {
    int h, i, j;
    cout << "Enter the size of array: ";
    cin >> h;
    int arr[h];
    for (i = 0; i < h; i++) {
        cout << "Enter arr[" << i << "] = ";
        cin >> arr[i];
    }

    for (i = 0; i < h; i++) {
        if (arr[0] < arr[i]) {
            arr[0] = arr[i];
        }
    }

    cout << "The greatest element in array = "
        << arr[0];

    for (i = 0; i < h; i++) {
        if (arr[0] > arr[i]) {
            arr[0] = arr[i];
        }
    }
}
```

```
cout << "\n The smallest element in array = "
      << arr[0];
```

```
return 0;
```

```
}
```

Output :-

```
Enter the size of array : 5
```

```
Enter arr[0] = 20
```

```
Enter arr[1] = 45
```

```
Enter arr[2] = 42
```

```
Enter arr[3] = 44
```

```
Enter arr[4] = 44
```

```
The greatest element in array = 20
```

```
The smallest element in array = 44
```

8. Write a program to find the GCD (Greatest common divisor) of the two positive integers using Euclidean algorithm.

⇒ program :-

```
#include <iostream>
using namespace std;

int gcd (int a, int b) {
    if (a == 0) {
        return b;
    }
    if (b == 0) {
        return a;
    }
    if (a == b) {
        return a;
    }
    if (a > b) {
        return (gcd (a-b, b));
    }
    else {
        return (gcd (a, b-a));
    }
}

int main () {
    int h1, h2;
```

```
cout << "Enter first number : ";
cin >> h1;

cout << "Enter second number : ";
cin >> h2;

cout << " GCD of " << h1 << " and " << h2 << " is "
      " << gcd(h1, h2);

return 0;
}
```

output :-

```
Enter first number : 360
Enter second number : 288
GCD of 360 and 288 is 72.
```

```
Enter first number : 54
Enter second number : 81
GCD of 54 and 81 is 27
```

g. Write a program to simulate the use of MOD operator.

⇒ Program:

```
#include<iostream>
using namespace std;
int main() {
    int num1, num2, mod;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    mod = num1 % num2;
    cout << num1 << " mod(%o) " << num2 << " = "
        << mod;
    return 0;
}
```

Output:

```
Enter first number: 5
Enter second number: 4
5 mod(%o) 4 = 1
```

Q. Write a program to perform the following operations in zero-one matrix.

- i. Disjunction
- ii. Conjunction

⇒ Program:

```
#include<iostream>
using namespace std;
int main(){
    int m, n, a, b, i, j;
    cout << "Enter the size of first matrix
    (rows and columns): ";
    cin >> m >> n;
    int first[m][n];
    cout << "Enter the elements of first matrix:
    \n";
    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            cout << "Enter a" << i+1 << j+1 << "=";
            cin >> first[i][j];
        }
    }
    cout << "Enter the elements of second matrix:
    \n";
    cout << "Enter the size of second matrix: ";
    cin >> a >> b;
```

```

for (i=0; i<q;
    int second [q][b];
cout << "Enter the elements of second matrix : \n";
for (i=0; i<q; i++) {
    for (j=0; j<b; j++) {
        cout << "Enter a " << i+1 << j+1 <<
            " = ";
        cin << second [i][j];
    }
}

// disjunction operation
int join [m][h];
if (m == q && h == b) {
    for (i=0; i<m; i++) {
        for (j=0; j<h; j++) {
            join [i][j] = first [i][j] || second
                [i][j];
        }
    }
}

cout << "The Disjunction operation on two matrix is \n";
for (i=0; i<m; i++) {
    for (j=0; j<h; j++) {
        cout << " " << join [i][j];
    }
    cout << "\n";
}

```

```

// conjunction operation
int meet[m][h];
if (m == a &amp; h == b) {
    for (i=0; i<m; i++) {
        for (j=0; j<h; j++) {
            meet[i][j] = first[i][j] &&
                           second[i][j];
        }
    }
}
cout << " \n conjunction operation on two
matrices: \n ";
for (i=0; i<m; i++) {
    for (j=0; j<h; j++) {
        cout << " " << meet[i][j];
    }
}
cout << "\n";
}

return 0;
}

```

Output:

Enter the size of first matrix (rows and columns)

: 3

3

Enter the elements of first matrix:

Enter $a_{11} = \frac{1}{1}$

Enter $a_{12} = 0$

Enter $a_{13} = 0$

Enter $a_{21} = 0$

Enter $a_{22} = 0$

Enter $a_{23} = \frac{1}{1}$

Enter $a_{31} = \frac{1}{1}$

Enter $a_{32} = \frac{1}{1}$

Enter $a_{33} = \frac{1}{1}$

Enter the size of second matrix (rows and columns) : 3

3

Enter the elements of second matrix:

Enter $a_{11} = \frac{1}{1}$

Enter $a_{12} = \frac{1}{1}$

Enter $a_{13} = 0$

Enter $a_{21} = 0$

Enter $a_{22} = 0$

Enter $a_{23} = 0$

Enter $a_{31} = 0$

Enter $a_{32} = 0$

Enter $a_{33} = \frac{1}{1}$

Disjunction operation on two matrices:

$\frac{1}{1} \quad \frac{1}{1} \quad 0$

$0 \quad 0 \quad \frac{1}{1}$

$\frac{1}{1} \quad \frac{1}{1} \quad \frac{1}{1}$

conjunction operation on two matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Q1. Write a program to find join & meet of a Boolean matrix.

→ Program:

```
#include<iostream>
using namespace std;

int main(){
    int m, n, a, b, i, j;

    cout << "Enter the size of first boolean matrix : ";
    cin >> m >> n;

    int first[m][n];

    cout << "Enter the elements of first boolean matrix : ";
    cout << endl;

    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            cout << "Enter a" << i+1 << j+1 << " = ";
            cin >> first[i][j];
        }
    }

    cout << "Enter the size of second boolean matrix : ";
    cin >> a >> b;

    int second[a][b];

    cout << "Enter the elements of second boolean matrix : ";
    for (i=0; i<a; i++) {
        for (j=0; j<b; j++) {
            cout << "Enter a" << i+1 << j+1 << " = ";
            cin >> second[i][j];
        }
    }
}
```

}

}

int join[m][h];

//calculating join of boolean matrix

if (m == a && h == b) {

 for (i = 0; i < m; i++) {

 for (j = 0; j < h; j++) {

 join[i][j] = first[i][j] || second[i][j];

 }

}

cout << "In the join operation between two matrices
: \h";

 for (i = 0; i < m; i++) {

 for (j = 0; j < h; j++) {

 cout << " " << join[i][j];

 }

 cout << "\h";

}

}

int meet[m][n];

//calculating meet of boolean matrix

if (m == a && n == b) {

 for (i = 0; i < m; i++) {

 for (j = 0; j < h; j++) {

 meet[i][j] = first[i][j] && second[i][j];

 }

}

}

```

cout << "\n The meet operation between two matrices : \n";
for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
        cout << " " << meet[i][j];
    }
    cout << "\n";
}
return 0;
}

```

Output:

Enter the size of first boolean matrix : 2

2

Enter the elements of first boolean matrix :

Enter $a_{11} = 1$

Enter $a_{12} = 0$

Enter $a_{21} = 0$

Enter $a_{22} = 1$

Enter the size of second boolean matrix : 2

2

Enter the elements of second boolean matrix :

Enter $a_{11} = 1$

Enter $a_{12} = 1$

Enter $a_{21} = 1$

Enter $a_{22} = 0$

The join operation between two matrices :

1 1

1 1

The meet operation between two matrices :

1 0

0 0