Bismillahhir Rahmanir Rahim

আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম

الجامعة الإسلامية العالمية شيتاغونغ

International Islamic University Chittagong

Assignment

Assignment No. – 02

Submission date- 17 September, 2021

Course Title- Data Structure (Theory)

Course Code: CSE-2322

Submited to-

Mohammed Shamsul Alam

Professor, Dept. of CSE, IIUC.

Cell: 01711941680, alam_cse@yahoo.com

Submitted by-

MD. SOROWAR MAHABUB RABBY

Matric ID: ***C201032***, Section: ***3AM*** , Semester: ***3rd***

Department of CSE (*Computer Science and Engineering*), IIUC

Cell: *01834756433, 01521564157, c201032@ugrad.iiuc.ac.bd*

Problem No. & Statement	*1. Write a program to interchange the row and column of a matrix.*

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

int main() {
    int a[10][10],
transpose[10][10], row, column,
i, j;

    cout << "Enter rows of
matrix: ";
    cin >> row;
    cout << "Enter columns of
matrix: ";
    cin >> column;
    cout << "\nEnter elements of
matrix:" << endl;


    for(int i= 1; i<=row; i++)
        for(int j= 1; j<=column;
j++)
            cin >> a[i][j];


    cout << "\n\nEntered
Matrix:\n" << endl;
    for(int i= 1; i<=row; i++)
    {
        for(int j= 1; j<=column;
j++)
            cout << a[i][j] << "
";
        cout << endl;
    }



    for(int i= 1; i<=row; i++)
        for(int j= 1; j<=column;
j++)
            transpose[j][i] =
a[i][j];


    cout << "\n\nTranspose of
Matrix:\n" << endl;
    for(int i= 1; i<=column;
i++)
    {
        for(int j= 1; j<=row;
j++)
            cout << " " <<
transpose[i][j];
        cout << endl;
    }

    return 0;
}
```

Problem No. & Statement	*2. Write a program to add two matrices.*

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

int main()
{
    int r, c, i, j;
    int a[132][132],
b[132][132], sum[132][132];

    cout << "Enter number of
rows: ";
    cin >> r;
    cout << "Enter number of
columns: ";
    cin >> c;

    cout << endl << "Enter
elements of 1st matrix:" <<
endl;
    for(i= 1; i<=r; i++)
        for(j= 1; j<=c; j++)
            cin >> a[i][j];

    cout << endl << "Enter
elements of 2nd matrix:" <<
endl;
    for(i= 1; i<=r; i++)
        for(j= 1; j<=c; j++)
            cin >> b[i][j];

    for(i= 1; i<=r; i++)
        for(j= 1; j<=c; j++)
            sum[i][j]= a[i][j] +
b[i][j];

    cout << endl << "Sum of two
matrix is:" << endl;
    for(i= 1; i<=r; i++)
    {
        for(j= 1; j<=c; j++)
```

```
            cout << sum[i][j] <<
"   ";
        cout << endl;
    }

    return 0;
}
```

Problem No. & Statement	*3. Write a program to calculate the rowsum and columnsum of a matrix.*

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

int main()
{
    int a[10][10], b[10][10],
mult[10][10], r1, c1, r2, c2, i,
j, k;

    cout << "Enter rows and
columns for matrix 1: ";
    cin >> r1 >> c1;
    cout << "Enter rows and
columns for matrix 2: ";
    cin >> r2 >> c2;

    if(c1!=r2)
        cout <<
"\n\n\t\tError!\nColumn of
matrix 1 not equal to row of
matrix 2.";
    else
    {
        cout << endl << "Enter
elements of matrix 1:" << endl;
        for(i= 1; i<=r1; i++)
            for(j= 1; j<=c1;
j++)
                cin >> a[i][j];

        cout << endl << "Enter
elements of matrix 2:" << endl;
        for(i= 1; i<=r2; i++)
            for(j= 1; j<=c2;
j++)
                cin >> b[i][j];

        for(i= 1; i<=r1; i++)
            for(j= 1; j<=c2;
j++)
                for(k= 1; k<=c1;
k++)
                    mult[i][j]+=
a[i][k] * b[k][j];

        cout << endl << "After
Multiplying, Matrix:" << endl;
        for(i= 1; i<=r1; i++)
        {
            for(j= 1; j<=c2;
j++)
                cout <<
mult[i][j] << " ";
            cout << endl;
        }
    }

    return 0;
}
```

Problem No. & Statement	*4. Write a program to calculate the multiplication of two matrices.*

```
#include <iostream>
#define size 132
using namespace std;

void rowSum(int
arr[size][size],int m,int n)
{
    int sum;
    for(int i= 1; i<=n; i++)
    {
        sum = 0;
        for(int j= 1; j<=m; j++)
            sum+= arr[i][j];

        cout << "Sum of row " <<
i << " is: " << sum << endl;
    }
}


void columnSum(int
arr[size][size], int m, int n)
{
    int sum;
    for(int i= 1; i<=n; i++)
    {
        sum = 0;
        for(int j= 1; j<=m; j++)
            sum+= arr[j][i];

        cout << "Sum of column "
<< i << " is: " << sum << endl;
```

```
    }
}

int main()
{
    int arr[size][size], m, n;

    cout << "Enter the size of
row: ";
    cin >> m;
    cout<<"\nEnter the size of
column: ";
    cin >> n;

    cout << "\nEnter the
matrix:" << endl;


    for(int i= 1; i<=m; i++)
        for(int j= 1; j<=n; j++)
            cin>>arr[i][j];

    cout << endl << endl;
    rowSum(arr, m, n);
    cout<<"\n\tAnd\n"<< endl;
    columnSum(arr, m,n);

    return 0;
}
```

Problem No. & Statement	*5. Write a program to check if a Matrix is a Sparse Matrix.*

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

int main()
{
    cout << "Enter row & column
of the matrix: ";
    int m, n;
    cin >> m >> n;
    int mat[m][n];
    int zeros= 0;

    cout << "Enter the elements
of the matrix:" << endl;
    for(int i= 1; i<=m; i++)
        for(int j= 1; j<=n; j++)
        {
            cin >> mat[i][j];
            if(mat[i][j]==0)
                zeros++;
        }

    if(zeros > (m*n)/2)
        cout << "The matrix is a
Sparse Matrix" << endl;
    else
        cout << "The matrix is
NOT a Sparse Matrix" << endl;

    return 0;

}
```

Problem No. & Statement	*6. Write a program to implement the push and pop operation of a stack*

```
#include<stdio.h>
#define SIZE 5

int Stack[SIZE+1], maxstk= 5,
top= 0, item;


int menu(void)
{
    int choice;
    do
    {
        printf("\n1-push\n2-
pop\n0-Exit\n");
        printf("Enter your
choice: ");
        scanf("%d",&choice);
        if(choice<0||choice>2)

printf("\nWrong...Choice
again...\n");
    }
    while(choice<0||choice>2);
    return (choice);
}

// Author: Sorowar Mahabub,  ID:
C201032



void push()
{
    printf("Enter the item: ");
```

```
    scanf("%d", &item);
    if(top==maxstk)
    {
        printf("OVERFLOW\n");
        return;
    }

    top= top+1;
    Stack[top]= item;
}

void pop()
{
    if(top==0)
    {
        printf("UNDERFLOW\n");
        return;
    }
    item= Stack[top];
    printf("\n%d is deleted!\n",
item);
    top= top-1;
}


void display()
{
    printf("Stored values in
Stack are: ");
    for(int i= 1; i<top; i++)
        printf("%d ", Stack[i]);
    printf("%d\n", Stack[top]);
}

int main()
{
    int choice;


    do
    {
        choice=menu();
        switch(choice)
        {
        case 1:
            push();
            display();
            break;
        case 2:
            pop();
            display();
            break;
        case 0:
            printf("End of
operation\n");
            break;
        }
    }
    while(choice!=0);
    return 0;
}
```

Problem No. & Statement	*7. Write a program to evaluate a Postfix expression.*

```
#include<stdio.h>

int stack[20];
int top = -1;
void push(int x)
{
    stack[++top] = x;
}
int pop()
{
    return stack[top--];
}
int main()
{
    char exp[20];
    char *p;
    int n1,n2,n3,num;
    printf("Enter the expression
:: ");
    scanf("%s",exp);
    p = exp;
    while(*p!= '\0')
    {
        if(isdigit(*p))
        {
            num = *p - 48;
            push(num);
        }
        else
        {
            n1 = pop();
            n2 = pop();
            switch(*p)
            {
            case '+':
            {
                n3 = n1 + n2;
                break;
            }
            case '-':
            {
                n3 = n2 - n1;
                break;
            }
            case '*':
```

```
            {
                n3 = n1 * n2;
                break;
            }
            case '/':
            {
                n3 = n2 / n1;
                break;
            }
            }
            push(n3);
        }
        p++;
    } // Author: Sorowar
Mahabub, C201032
    printf("The result of
expression %s = %d",exp,pop());
    return 0;
}
```

Problem No. & Statement	*8. Write a program to convert an Infix expression into its equivalent Postfix expression.*

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

char stack[100];
int top = -1;
void push(char x)
{
    stack[++top] = x;
}
char pop()
{
    if(top == -1)
        return -1;
    else
        return stack[top--];
}
int priority(char x)
{
    if(x == '(')
        return 0;
    if(x == '+' || x == '-')
        return 1;
    if(x == '*' || x == '/')
        return 2;
    return 0;
}
int main()
{
    char exp[100];
    char *e, x;
    printf("Enter the expression
: ");
    scanf("%s",exp);
    printf("\n");
    e = exp;
    while(*e != '\0')
    {
        if(isalnum(*e))
            printf("%c ",*e);
        else if(*e == '(')
            push(*e);
        else if(*e == ')')
        {
            while((x = pop()) !=
'(')
                printf("%c ",
x);
        }
        else // Author: Sorowar
Mahabub, C201032
        {

while(priority(stack[top]) >=
priority(*e))
                printf("%c
",pop());
            push(*e);
        }
        e++;
    }
    while(top != -1)
    {
        printf("%c ",pop());
    }
    return 0;
}
```

Problem No. & Statement	*9. a) Find the length of a string S*

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

int main()
{
    string s;
    cin >> s;
    int length= 0;
    for(int i= 0; s[i]!='\0';
i++)
        length++;
    cout << "The length of  " <<
s << " is " << length << endl;

    return 0;
}
```

Problem No. & Statement	9. b) Copy string S2 to S1.

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

int main()
{
    char s1[100], s2[100], i;
    cout << "Enter String S1: ";
    cin >> s1;

    for(i=0; s1[i]!='\0'; i++)
        s2[i]=s1[i];

    s2[i]='\0';

    cout << "\nCopied String S2
is : " << s2;

    return 0;
}
```

Problem No. & Statement	9. . c) Concatenate string S2 to S1.

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

int main()
{
    char str1[100], str2[100];
    int i = 0, j = 0;

    cout << "Enter first
string:";
    cin >> str1;

    cout << "Enter second
string:";
    cin >> str2;

    while(str1[i] != '\0')
        i++;

    while(str2[j] != '\0')
    {
        str1[i] = str2[j];
        j++;
        i++;
    }
    str1[i] = '\0';

    cout << "Concatenated
string:" << str1;

    return 0;
}
```

Problem No. & Statement	9. d) Compare two strings S1 and S2

```
#include<bits/stdc++.h>
#include<iostream>
#include<string.h>
using namespace std;

int main()
{
    char
str1[50],str2[50],i=0,j=0,flag=0
;

    cout<<"Enter first string ::
";
    gets(str1);

    cout<<"\nEnter Second string
:: ";
    gets(str2);

    while(str1[i]!='\0')
        i++;

    while(str2[j]!='\0')
        j++;
```

```
    if(i!=j)
        flag=0;

    else
        for(i=0,j=0;
str1[i]!='\0',str2[j]!='\0';
i++,j++)
            if(str1[i]==str2[j])
                flag=1;

    if(flag==0)
        cout<<"\nStrings are not
equal.\n";
    else
        cout<<"\nStrings are
equal.\n";

    return 0;
}
```

Problem No. & Statement	*9. e) Reverse a string S.*

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

using namespace std;

int main()
{
    char s1[132];
    int n= 0, i= 0;

    cout << "Enter the String:
";
    cin >> s1;

    while(s1[i]!='\0')
        i++;

    n= i;
    char s2[i];

    i = 0;
    while (i != n + 1)
    {
        s2[i]= s1[n-i-1];
        i++;
    }

    cout << "\nReverse of the
entered string \"" << s1 << "\"
is : \"" << s2 << "\"\n";

    return 0;
}
```

Problem No. & Statement	*10. Write a program to insert a string S into a text T so that S begins in position K of T.*

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    char str[1032], add[332];
    int k;

// Author: Sorowar Mahabub
    cout << "Enter String: ";
    gets(str);

    cout << endl << "Enter the
letrer/ word: ";
    cin >> add;

    cout << endl << "Enter
potion: " << endl;
    cin >> k;

    int l1= strlen(str);
    int l2= strlen(add);
    int tl= l1+l2;
    int i, j;
    for(i= tl, j= l1-1; i>k; i--
, j--)
        str[i]= str[j];

    for(i= k, j= 0; j<l2; i++,
j++)
        str[i]= add[j];

    str[i]= ' ';

    for(i= 0; i<=tl; i++)
        cout << str[i];
    cout << endl;

    return 0;
}
```

Problem No. & Statement	*11. A text T in memory. Write a program to delete a*

	string S of length L from Kth position in T.

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    char str[1032], p[332],
q[332];
    //Author: Sorowar Mahabub
    cout << "Enter String: ";
    gets(str);

    cout << "Enter P: ";
    cin >> p;

    int k;
    cout << "Enter Position: ";
    cin >> k;

    int l1= strlen(str);
    int lp= strlen(p);

    int i, j;
    for(i= k-1, j= (k+lp-1);
j<l1; i++, j++)
        str[i]= str[j];


    for(i= 0; i<=(l1-lp-1); i++)
        cout << str[i];
    cout << endl;

    return 0;
}
```

Problem No. & Statement	***12. A text T and patterns P and Q in memory in memory. Write a program to replace the first occurrence of a pattern (P) in T by Q.***

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int l, l1, l2, i, j, k;
    cout << "Main string length:
";
    cin >> l;
    cout<<"Pattern string 1
length: ";
    cin >> l1;
    cout<<"Pattern string 2
length: ";
    cin >> l2;

    char t[l+l1+l2+32],
p[l1+32], q[l2+32];
    cout << "Main string: ";
    for(i= 1; i<=l; i++)
        cin >> t[i];

    cout<<"1st pattern: ";
    for(i= 1; i<=l1; i++)
        cin >> p[i];

    cout << "2nd pattern: ";
    for(i= 1; i<=l2; i++)
        cin >> q[i];

    for(i= 1; i<=l-l1+1; i++)
    {
        int pos= i;
        int flag= 0;
        for(j= i, k= 1; j<i+l1;
j++, k++)
            if(t[j]!=p[k])
            {
                flag= 1;
                pos= 0;
                break;
            }

        if(flag==0)
        {


// Pattern P Substructing
            for(j= pos+l1; j<=l;
j++)
                t[j-l1]= t[j];

            l-= l1;
            l+= l2;
            // Inserting pattern
Q
            for(j= pos+l2; j<=l;
j++)
                t[j]= t[j-l2];

            for(j= pos, k= 1;
j<pos+l2; j++, k++)
                t[j]= q[k];

            break;
```

```
        }
    }

    cout << "Updated string: ";
    for(i= 1; i<=l; i++)
        cout << t[i];

    return 0;
}
```

Problem No. & Statement	*13. Write a program that will read a string (S) and find the index of the first occurrence of a pattern (P) in the string S.*

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int l1, l2, cnt, i, j, f= 1;
    char str1[10032],
str2[1032];
    //Author: Sorowar Mahabub
(C201032)
    cout << "Enter the length of
the Main String: ";
    cin >> l1;

    cout << "Enter the Main
String: ";
    for(i= 1; i<=l1; i++)
        cin >> str1[i];


    cout << "Enter the length of
the Pattern String: ";
    cin >> l2;

    cout << "Enter the 2nd
String: ";
    for(i= 1; i<=l2; i++)
        cin >> str2[i];


    for(i= 1; i<=(l1-l2+1); i++)
    {
        cnt = 0;
        for(j= 0; j<l2; j++)
            if(str1[i+j] ==
str2[j+1])
                cnt++;

        if(cnt==l2)
        {
            cout << "Pattern
String found & 1st index is " <<
i << "." << endl;
            f= 0;
            break;
        }
    }

    if(f!=0)
        cout << "Pattern String
Not Found!" << endl;


    return 0;
}
```

Problem No. & Statement	*14. Write a program which calculates the no. of occurrence of each letter of an input text.*

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

int main()
{
    int ip, len, a, b, c, d, e, f,
g, h, i, j, k, l, m, n;
    int o, p, q, r, ss, t, u, v, w,
x, y, z, mm;
    char s[1032];

a=b=c=d=e=f=g=h=i=j=k=l=m=n=o=p=q=r
= 0;
    ss=t=u=v=w=x=y=z= 0;
    printf("Enter the string: ");
    gets(s);
    len = strlen(s);
    for(ip=0; ip<=(len-1); ip++)
    {
        if(s[ip]=='a')
            a++;
        else if(s[ip]=='b')
            b++;
        else if(s[ip]=='c')
            c++;
        else if(s[ip]=='d')
            d++;
        else if(s[ip]=='e')
            e++;
```

```
        else if(s[ip]=='f')
            f++;
        else if(s[ip]=='g')
            g++;
        else if(s[ip]=='h')
            h++;
        else if(s[ip]=='i')
            i++;
        else if(s[ip]=='j')
            j++;
        else if(s[ip]=='k')
            k++;
        else if(s[ip]=='l')
            l++;
        else if(s[ip]=='m')
            m++;
        else if(s[ip]=='n')
            n++;
        else if(s[ip]=='o')
            o++;
        else if(s[ip]=='p')
            p++;
        else if(s[ip]=='q')
            q++;
        else if(s[ip]=='r')
            r++;
        else if(s[ip]=='s')
            ss++;
        else if(s[ip]=='t')
            t++;
        else if(s[ip]=='u')
            u++;
        else if(s[ip]=='v')
            v++;
        else if(s[ip]=='w')
            w++;
        else if(s[ip]=='x')
            x++;
        else if(s[ip]=='y')
            y++;
        else if(s[ip]=='z')
            z++;
    }


    if(a!=0)
        printf("a : %d\n", a);
    if(b!=0)
        printf("b : %d\n", b);
    if(c!=0)
        printf("c : %d\n", c);
    if(d!=0)
        printf("d : %d\n", d);
    if(e!=0)
        printf("e : %d\n", e);
    if(f!=0)
        printf("f : %d\n", f);
    if(g!=0)
        printf("g : %d\n", g);
    if(h!=0)
        printf("h : %d\n", h);
    if(i!=0)
        printf("i : %d\n", i);
    if(j!=0)
        printf("j : %d\n", j);
    if(k!=0)
        printf("k : %d\n", k);
    if(l!=0)
        printf("l : %d\n", l);
    if(m!=0)
        printf("m : %d\n", m);
    if(n!=0)
        printf("n : %d\n", n);
    if(o!=0)
        printf("o : %d\n", o);
    if(p!=0)
        printf("p : %d\n", p);
    if(q!=0)
        printf("q : %d\n", q);
    if(r!=0)
        printf("r : %d\n", r);
    if(ss!=0)
        printf("s : %d\n", ss);
    if(t!=0)
        printf("t : %d\n", t);
    if(u!=0)
        printf("u : %d\n", u);
    if(v!=0)
        printf("v : %d\n", v);
    if(w!=0)
        printf("w : %d\n", w);
    if(x!=0)
        printf("x : %d\n", x);
    if(y!=0)
        printf("y : %d\n", y);
    if(z!=0)
        printf("z : %d\n", z);

    return 0;
}
```

Problem No. & Statement	***15. Write a program that will read a positive integer in base b (2 <= b <= 16) and convert it into base d (2 <= d <= 16).***

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    string s;
    int b, d;
    cout << "Enter number to
convert: ";
    cin >> s;
```

```
    cout << "Enter base of the
above number: ";
    cin >> b;
    cout << "Enter base to
convert: ";
    cin >> d;

    //Any base to decimal
    long long int dec= 0, val=
1;
    for(int i=(int)s.size()-1;
i>=0; i--)
    {
        int temp;
        if(s[i]>='0' &&
s[i]<='9')
            temp= (int)s[i]-'0';

        else
            temp= (int)s[i]-
'A'+10;

        dec+= (temp*val);
        val*= b;
    }

    //Decimal to any base
    int ans[100]= {};
    int j= 0;
    while(dec)
    {
        ans[j++]= dec%d;
        dec/= d;
    }

    int l= j;
    for(int i= l-1; i>=0; i--)
    {
        if(ans[i]>=10)
            cout <<
(char)(ans[i]-10+'A');
        else
            cout << ans[i];
    }

    return 0;
}
```

Problem No. & Statement	*16. Write a program to determine the Greatest Common Divisor (GCD) & Least Common Multiple (LCM) of two given positive integers.*

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


int main()
{
    int num1, num2, gcd, lcm,
rem, numerator, denominator;

    cout << "Enter two numbers:
";
    cin >> num1 >> num2;

    if(num1>num2)
    {
        numerator = num1;
        denominator = num2;
    }
    else
    {
        numerator = num2;
        denominator = num1;
    }

    rem= numerator %
denominator;
    while(rem != 0)
    {
        numerator= denominator;
        denominator= rem;
        rem= numerator %
denominator;
    }

    gcd= denominator;
    lcm= (num1*num2) / gcd;
    cout << "GCD = " << gcd << "
and LCM= " << lcm << endl;

    return 0;
}
```