ASSIGNMENT-LAB 02

Course Code: CSE - 2322 **Course Title:** Data Structures (& Lab)
Course Teacher: Mohammed Shamsul Alam

Name: Ezaz Ahmed **ID:** C223009 **Section:** 3AM

Problem 01: **Write a program to interchange the row and column of a matrix.**

Answer:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   int r, c;
   cout << "Enter the number of rows and columns: ";
   cin >> r >> c;
   int A[r + 1][c + 1];
   cout << "Enter the elements of the matrix:" << endl;
   for (int i = 1; i <= r; i++)
   {
      for (int j = 1; j <= c; j++)
      {
         cin >> A[i][j];
      }
   }
   cout << "\nOriginal Matrix:" << endl;
   for (int i = 1; i <= r; i++)
   {
      for (int j = 1; j <= c; j++)
      {
         cout << A[i][j] << " ";
      }
      cout << endl;
   }
   int B[r + 1][c + 1];
   for (int i = 1; i <= r; i++)
   {
      for (int j = 1; j <= c; j++)
      {
            B[j][i] = A[i][j];
        }
    }
    cout <<"\nInterchanged Matrix:" << endl;
    for (int i = 1; i <= c; i++)
    {
        for (int j = 1; j <= r; j++)
        {
            cout << B[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Problem 02: **Write a program to add two matrices.**

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int r, c;
    cout << "Enter the number of rows and columns: ";
    cin >> r >> c;
    int A[r + 1][c + 1];
    cout << "Enter the elements of Matrix A:" << endl;
    for (int i = 1; i <= r; i++)
    {
        for (int j = 1; j <= c; j++)
        {
            cin >> A[i][j];
        }
    }
    int B[r + 1][c + 1];
    cout << "Enter the elements of Matrix B:" << endl;
    for (int i = 1; i <= r; i++)
    {
        for (int j = 1; j <= c; j++)
        {
            cin >> B[i][j];
        }
    }
    int C[r + 1][c + 1];
    for (int i = 1; i <= r; i++)
    {
        for (int j = 1; j <= c; j++)
        {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
    cout << "After Adding:" << endl;
    for (int i = 1; i <= r; i++)
    {
        for (int j = 1; j <= c; j++)
        {
            cout << C[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Problem 03: **Write a program to calculate the rowsum and columnsum of a matrix.**

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int r, c;
    cout << "Enter the number of rows: ";
    cin >> r;
    cout << "Enter the number of columns: ";
    cin >> c;
    int mat[r+1][c+1];
    cout << "Enter the elements of the matrix:" << endl;
    for (int i = 1; i <= r; i++)
    {
        for (int j = 1; j <= c; j++)
        {
            cin >> mat[i][j];
        }
    }
    int rS, cS;
    cout << "Row sums:" << endl;
    for (int i = 1; i <= r; i++)
    {
        rS = 0;
        for (int j = 1; j <= c; j++)
        {
            rS += mat[i][j];
        }
        cout << "Row " << i << ": " << rS << endl;
    }
    cout << "Column sums:" << endl;
    for (int j = 1; j <= c; j++)
    {
        cS = 0;
        for (int i = 1; i <= r; i++)
        {
            cS += mat[i][j];
        }
        cout << "Column " << j << ": " << cS << endl;
    }
    return 0;
}
```

Problem 04: Write a program to calculate the multiplication of two matrices.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int r1, c1, r2, c2;
```

```
cout << "Enter the number of rows and columns for Matrix A: ";
cin >> r1 >> c1;
cout << "Enter the number of rows and columns for Matrix B: ";
cin >> r2 >> c2;
if (c1 != r2)
{
   cout <<"Matrix multiplication is not possible."<< endl;
   return 0;
}
int A[r1][c1];
int B[r2][c2];
int C[r1][c2];
cout << "Enter the elements of Matrix A:" << endl;
for (int i = 0; i < r1; i++)
{
   for (int j = 0; j < c1; j++)
   {
      cin >> A[i][j];
   }
}
cout << "Enter the elements of Matrix B:" << endl;
for (int i = 0; i < r2; i++)
{
   for (int j = 0; j < c2; j++)
   {
      cin >> B[i][j];
   }
}
for (int i = 0; i < r1; i++)
{
   for (int j = 0; j < c2; j++)
   {
      C[i][j] = 0;
      for (int k = 0; k < c1; k++)
      {
         C[i][j] += A[i][k] * B[k][j];
      }
```

```
        }
    }
    cout << "After Multiplication:" << endl;
    for (int i = 0; i < r1; i++)
    {
        for (int j = 0; j < c2; j++)
        {
            cout << C[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Problem 05: **Write a program to check if a Matrix is a Sparse Matrix.**

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int r, c;
    cout << "Enter the number of rows: ";
    cin >> r;
    cout << "Enter the number of columns: ";
    cin >> c;
    int mat[r][c];
    int z=0;
    cout << "Enter the elements of the matrix:" << endl;
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            cin >> mat[i][j];
            if (mat[i][j] == 0)
            {
                z++;
            }
        }
```

```
   }
   int t=r*c;
   int th= t / 2;

   if (z > th)
   {
      cout << "The matrix is a sparse matrix." << endl;
   }
   else
   {
      cout << "The matrix is not a sparse matrix." << endl;
   }

   return 0;
}
```

Problem 06: **Write a program to implement the push and pop operation of a stack.**

Answer:

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

#define SIZE 5

int stack[SIZE + 1], top = 0;

int menu(void)
{
   int choice;
   do
   {
      cout << "\n1-Push\n2-Pop\n0-Exit\n";
      cout << "Enter your choice: ";
      cin >> choice;

      if (choice < 0 || choice > 2)
      {
         cout << "\nWrong choice. Please choose again.\n";
```

```
        }
    }
    while (choice < 0 || choice > 2);
    return choice;
}

void push(int value)
{
    if (top >= SIZE)
    {
        cout << "Stack is full. Cannot push more elements.\n";
        return;
    }
    stack[top] = value;
    top++;
}

void pop()
{
    if (top <= 0)
    {
        cout << "Stack is empty. Cannot pop.\n";
        return;
    }
    top--;
}

void display()
{
    if (top <= 0)
    {
        cout << "Stack is empty.\n";
        return;
    }
    cout << "Stack contents: ";
    for (int i = 0; i < top; i++)
    {
        cout << stack[i] << " ";
    }
    cout << "\n";
}

int main()
{
    int choice;
    do
    {
        choice = menu();
        switch (choice)
        {
        case 1:
        {
            int value;
            cout << "Enter the value to push: ";
            cin >> value;
            push(value);
            display();
        }
        break;
        case 2:
            pop();
            display();
            break;
        case 0:
            cout << "End of stack operations.\n";
            break;
        }
    }
    while (choice != 0);
    return 0;
}
```

Problem 07: **Write a program to evaluate a Postfix expression.**

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    stack<int> S;
    string s;
    cin >> s;
    for (char c : s)
    {
        if (isdigit(c))
        {
            S.push(c - '0');
        }
        else if (c == '+' || c == '-' || c == '*' || c == '/' || c == '^')
        {
            int o2 = S.top();
            S.pop();
            int o1 = S.top();
            S.pop();

            int res;
            if (c == '+')
            {
                res = o1 + o2;
            }
            else if (c == '-')
            {
                res = o1 - o2;
            }
            else if (c == '*')
            {
                res = o1 * o2;
            }
            else if (c == '/')
            {
                if (o2 != 0)
                {
                    res = o1 / o2;
                }
                else
                {
                    cerr << "Error: Division by zero." << endl;
                    exit(1);
                }
            }
            else if (c == '^')
            {
                res = pow(o1, o2);
            }
            S.push(res);
        }
    }

    int fr = S.top();
    cout << "Result: " << fr << endl;

    return 0;
}
```

Problem 08: **Write a program to convert an Infix expression into its equivalent Postfix expression.**

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    stack<char> S;
    string in, po = "";
    cout << "Enter an Infix expression: ";
    cin >> in;
    for (char c : in)
    {
        if (c >= '0' && c <= '9' || c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z')
        {
            po += c;
        }
        else if (c == '(')
        {
            S.push(c);
        }
        else if (c == ')')
        {
            while (!S.empty() && S.top() != '(')
            {
                po += S.top();
                S.pop();
            }
            S.pop();
        }
        else
        {
            while (!S.empty() && (S.top() == '*' || S.top() == '/' || S.top() == '+' ||
S.top() == '-' || S.top() == '^'))
            {
                po += S.top();
                S.pop();
            }
            S.push(c);
        }
    }

    while (!S.empty())
    {
        po += S.top();
        S.pop();
    }
    cout << "Equivalent Postfix expression: " << po << endl;
    return 0;
}
```

Problem 09: Write a program to implement the following string operation without using any built in functions related to string. a) Find the length of a string S b) Copy string S2 to S1. c) Concatenate string S2 to S1. d) Compare two strings S1 and S2 e) Reverse a string S.

Answer:

a)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   string S1;
   cout << "Enter a string S1: ";
   getline(cin,S1);
   int i = 0;
   while (S1[i] != '\0')
   {
      i++;
   }
   cout << "Length of S1: " << i << endl;
   return 0;
}
```

b)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   char S1[100], S2[100];
   cout << "Enter string S1: ";
   cin >> S1;
   cout << "Enter string S2: ";
   cin >> S2;
   int i = 0;
   while (S2[i] != '\0')
   {
      S1[i] = S2[i];
      i++;
   }
   S1[i] = '\0';
   cout << "After Copy: " << S1 << endl;
   return 0;
}
```

c)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   char S1[100], S2[100];
   cout << "Enter string S1: ";
   cin >> S1;
   cout << "Enter string S2: ";
   cin >> S2;
   int j = 0;
   while (S1[j] != '\0')
   {
      j++;
   }
   int i = 0;
   while (S2[i] != '\0')
   {
      S1[j] = S2[i];
      j++;
      i++;
   }
   S1[j] = '\0';
   cout << "After Concatenation: " << S1 << endl;
   return 0;
}
```

d)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    char S1[100], S2[100];
    cout << "Enter string S1: ";
    cin >> S1;
    cout << "Enter string S2: ";
    cin >> S2;
    int i = 0;
    while (S1[i] != '\0' && S2[i] != '\0' &&S1[i] == S2[i])
    {
        i++;
    }
    if (S1[i] == '\0' && S2[i] == '\0')
    {
        cout << "Strings are equal" << endl;
    }
    else
    {
        cout << "Strings are not equal" << endl;
    }
    return 0;
}
```

e)

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string S;
    cout << "Enter a string: ";
    getline(cin, S);
    int len = 0;
    while(S[len] != '\0')
    {
        len++;
    }
    for (int i = 0; i < len / 2; i++)
    {
        char te = S[i];
        S[i] = S[len - i - 1];
        S[len - i - 1] = te;
    }
    cout << "Reversed string: " << S << endl;
    return 0;
}
```

Problem 10: Write a program to insert a string S into a text T so that S begins in position K of T.

Answer:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    cout << "Enter the original text: ";
    string S;
    getline(cin,S);
    cout << "Enter the string to insert: ";
    string T;
    getline(cin,T);
    cout << "Enter the position to insert at: ";
    int K;
    cin >>K;
    if (K < 1 || K > S.length() + 1)
    {
        cout << "Invalid position." << endl;
        return 0;
    }
    string SI = "";
    for (int i = 0; i < K-1; i++)
    {
        SI += S[i];
    }
    SI+=T;
    for (int i = K-1; i < S.length(); i++)
    {
        SI += S[i];
    }
    cout << "After insertion: " << SI << endl;
    return 0;
}
```

Problem 11: A text T in memory. Write a program to delete a string S of length L from Kth position in T.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    cout << "Enter the original text: ";
    string T;
    getline(cin, T);
    cout << "Enter the position to delete from: ";
    int K;
    cin >> K;
    if (K < 1 || K > T.length())
    {
        cout << "Invalid position" << endl;
        return 0;
    }
    cout << "Enter the length of string to delete: ";
    int L;
    cin >> L;
    if (L < 1 || L > T.length() - K + 1)
    {
        cout << "Invalid length entered!" << endl;
        return 0;
    }
    string SN = "";
    for (int i = 0; i < K - 1; i++)
    {
        SN += T[i];
    }
    for (int i = K - 1 + L; i < T.length(); i++)
    {
        SN += T[i];
    }

    cout << "After deletion: " << SN << endl;

    return 0;
}
```

Problem 12: 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.

Answer:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string S, P;
    cout << "Enter the string: ";
    getline(cin, S);
    cout << "Enter the pattern: ";
    getline(cin, P);
    int a = S.length();
    int b = P.length();
    int ind = -1;
    for (int i = 0; i <= a - b; i++)
    {
        bool found = true;
        for (int j = 0; j < b; j++)
        {
            if (S[i + j] != P[j])
            {
                found = false;
                break;
            }
        }
        if (found)
        {
            ind = i + 1;
            break;
        }
    }

    if (ind != -1)
    {
        cout << "Pattern found at index: " << ind << endl;
    }
    else
    {
        cout << "Pattern not found." << endl;
    }

    return 0;
}
```

Problem 13: **A text T and patterns P and Q are given. Write a program to replace the first occurrence of a pattern (P) in T by Q.**

Answer:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string T, P, Q;
    cout << "Enter the string: ";
    getline(cin, T);
    cout << "Enter the pattern to be replaced: ";
    getline(cin, P);
    cout << "Enter the replacement pattern: ";
    getline(cin, Q);

    int a = T.length();
    int b = P.length();
    int ind = -1;
```

```
for (int i = 0; i <= a - b; i++)
{
    bool found = true;
    for (int j = 0; j < b; j++)
    {
        if (T[i + j] != P[j])
        {
            found = false;
            break;
        }
    }
    if (found)
    {
        ind = i + 1;
        break;
    }
}

if (ind != -1)
{
    string SN = "";
    for (int i = 0; i < ind - 1; i++)
    {
        SN += T[i];
    }
    SN += Q;
    for (int i = ind - 1 + b; i < a; i++)
    {
        SN += T[i];
    }
    cout << "After replacement: " << SN << endl;
}
else
{
    cout << "Pattern not found in the text." << endl;
}
return 0;
```

```
}
```

Problem 14: Write a program which calculates the no. of occurrence of each letter of an input text.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   string s;
   cout << "Enter the string: ";
   getline(cin, s);
   int lc[26] = {0};

   for (char b : s)
   {
      if (b >= 'a' && b <= 'z')
      {
         lc[b - 'a']++;
      }
      if (b >= 'A' && b <= 'Z')
      {
         b += 'a' - 'A';
         lc[b - 'a']++;
      }
   }
   cout << "Letter occurrences:" << endl;
   for (int i = 0; i < 26; i++)
   {
      if (lc[i] > 0)
      {
         cout << char('a' + i) << ": " << lc[i] << " occurrences" << endl;
      }
   }
   return 0;
}
```

<u>Problem 15:</u> Write a program that will read a positive integer in base b (2 <= b <= 16) and convert it into base d (2 <= d <= 16).

<u>Answer:</u>

```
#include<bits/stdc++.h>
using namespace std;
char intToChar(int k)
{
   if (k >= 0 && k <= 9)
   {
      return char(k + '0');
   }
   else if (k >= 10 && k <= 15)
   {
      return char(k - 10 + 'A');
   }
   return 'X';
}

int charToInt(char c)
{
   if (c >= '0' && c <= '9')
   {
      return c - '0';
   }
   else if (c >= 'A' && c <= 'F')
   {
      return 10 + (c - 'A');
   }
   else if (c >= 'a' && c <= 'f')
   {
      return 10 + (c - 'a');
   }
   return -1;
}

int main()
{
   int b, d;
i_b:
   cout << "Enter base (2 <= b <= 16): ";
   cin >> b;
   if (b < 2 || b > 16)
   {
      cout << "Invalid base." << endl;
      goto i_b;
   }
   cout << "Enter number in base " << b << ": ";
   string n;
   cin >> n;
i_d:
   cout << "Enter convert base (2 <= d <= 16): ";
   cin >> d;

   if (d < 2 || d > 16)
   {
      cout << "Invalid base." << endl;
      goto i_d;
   }

   int n10 = 0;
   int p = 0;
   int length = n.length();
   for (int i = length - 1; i >= 0; --i)
   {
      n10 += charToInt(n[i]) * pow(b, p);
      p++;
   }

   vector<char> dig;
   while (n10 > 0)
   {
      dig.push_back(intToChar(n10 % d));
      n10 /= d;
   }
   cout << "Num " << n << " in base " << d << " is: ";
   for (int i = dig.size() - 1; i >= 0; i--)
   {
      cout << dig[i];
   }
   cout << endl;

   return 0;
}
```

Problem 15: Write a program to determine the Greatest Common Divisor (GCD) & Least Common Multiple (LCM) of two given positive integers.

Answer:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a, b;
    cout << "Enter the first positive integer: ";
    cin >> a;
    cout << "Enter the second positive integer: ";
    cin >> b;
    int c = a, d = b;
    while (d != 0)
    {
        int tem = d;
        d = c % d;
        c = tem;
    }
    int gcd = c;
    int lcm = (a * b) / gcd;
    cout << "GCD of " << a << " and " << b << " is: " << gcd << endl;
    cout << "LCM of " << a << " and " << b << " is: " << lcm << endl;
    return 0;
}
```