

Ans to the Question Number – 1

interchange the row and column of a matrix.

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

int main()
{
    int row, col;
    cout<<"Row and Column: ";
    cin >> row >> col;
    int arr[row][col];
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            cin >> arr[i][j];
        }
    }
    for (int i = 1; i <= col; i++)
    {
        {
            for (int j = 1; j <= row; j++)
            {
                cout << arr[j][i] << " ";
            }
            cout << endl;
        }
    }
    return 0;
}
```

Ans to the Question Number – 2

add two matrices.

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

int main()
{
    int row, col;
    cout<<"Row and Column : ";
    cin >> row >> col;

    int arr1[row][col], arr2[row][col],
sum[row][col];

    cout<<"Enter 1st Matrix:\n";

    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            cin >> arr1[i][j];
        }
    }
    cout<<"Enter 2nd Matrix:\n";
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            cin >> arr2[i][j];
        }
    }

    cout<<"Sum of 2 Matrix:\n";
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            sum[i][j] = arr1[i][j] + arr2[i][j];
        }
    }
}
```

```

        cout << sum[i][j] << " ";
    }
    cout << endl;
}
return 0;
}

```

Ans to the Question Number – 3 **calculate the rowsum and columnsum of a matrix.**

```

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

int main()
{
    int row, col;
    cin >> row >> col;
    int arr[row][col];
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            cin >> arr[i][j];
        }
    }
    int RowSum = 0, ColSum = 0;
    for (int i = 1; i <= row; i++)
    {
        RowSum = 0;
        for (int j = 1; j <= col; j++)
        {
            RowSum += arr[i][j];
        }
        cout << "Sum of row " << i << " = " <<
RowSum << endl;
    }
    cout << endl;
    for (int i = 1; i <= col; i++)
    {
        ColSum = 0;

```

```

        for (int j = 1; j <= row; j++)
        {
            ColSum += arr[j][i];
        }
        cout << "Sum of column " << i << " = "
<< ColSum << endl;

    }
    return 0;
}

```

Ans to the Question Number – 4 **calculate the multiplication of two matrices.**

```

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

int main()
{
    int row1, row2, col1, col2;

    cout << "Row and Column of 1st Matrix: ";
    cin >> row1 >> col1;
    int arr1[row1][col1];
    for (int i = 0; i < row1; i++)
    {
        for (int j = 0; j < col1; j++)
        {
            cin >> arr1[i][j];
        }
    }

    cout << "Row and Column of 2nd Matrix: ";
    cin >> row2 >> col2;
    int arr2[row2][col2];
    for (int i = 0; i < row2; i++)
    {
        for (int j = 0; j < col2; j++)
        {

```

```

        cin >> arr2[i][j];
    }
}

int multi[row1][col2];
if (row2 != col1)
{

    cout << "\nMultiplication is not
Possible!" << endl;
}

else
{
    for (int i = 0; i < row1; i++)
    {
        for (int j = 0; j < col2; j++)
        {
            multi[i][j] = 0;
            for (int k = 0; k < col2; k++)
            {
                multi[i][j] += arr1[i][k] *
arr2[k][j];
            }
        }
    }
    cout << "Multiplication is:\n" << endl;
    for (int i = 0; i < row1; i++)
    {
        for (int j = 0; j < col2; j++)
        {
            cout << multi[i][j] << " ";
        }
        cout << endl;
    }
}
return 0;
}

```

Ans to the Question Number – 5

check if a Matrix is a Sparse Matrix.

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

int main()
{
    int row, col;
    cout << "Row and Column : ";
    cin >> row >> col;
    int arr[row][col];
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            cin >> arr[i][j];
        }
    }
    int cnt = 0;
    for (int i = 1; i <= row; i++)
    {
        for (int j = 1; j <= col; j++)
        {
            if (arr[i][j] == 0)
            {
                cnt++;
            }
        }
    }
    if (cnt > ((col * row) / 2))
    {
        cout << "The Matrix is a Sparse Matrix"
<< endl;
    }
    else
    {

```

```

        cout << "The Matrix is not a Sparse
Matrix" << endl;
    }

    return 0;
}

```

Ans to the Question Number – 6 **implement the push and pop operation of a stack**

```

#include<bits/stdc++.h>
#define SIZE 5

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

int menu(void)
{
    int choice;
    do
    {
        printf("1-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);
}

void push()
{
    if (top == SIZE)
    {
        printf("Stack Overflow\n");
    }
}

```

```
    }
    else
    {
        printf("Enter a value to push : ");
        int item;
        scanf("%d", &item);

        top++;
        stack[top] = item;
    }
}
void pop()
{
    if (top == 0)
    {
        printf("Stack Underflow\n");
    }
    else
    {
        int item;
        item = stack[top];
        top--;
    }
}

void display()
{
    if (top == 0)
    {
        printf("The Stack is Empty\n");
    }
    else
    {
        printf("The Stack elements are : ");
        for (int i = 1; i <= top; i++)
        {
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
}
```



```

    }

}

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;
}

```

Ans to the Question Number – 7 **evaluate a Postfix expression.**

```

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

int main()
{
    stack<string>s;

```

```
cout << "Postfix expression : " << endl;
while (1)
{
    string str;
    cin >> str;
    s.push(str);

    if (s.top() == "+")
    {
        s.pop();
        int a = stoi(s.top());
        s.pop();
        int b = stoi(s.top());
        s.pop();

        string ans = to_string(b + a);
        s.push(ans);
    }
    else if (s.top() == "-")
    {
        s.pop();
        int a = stoi(s.top());
        s.pop();
        int b = stoi(s.top());
        s.pop();

        string ans = to_string(b - a);
        s.push(ans);
    }

    else if (s.top() == "*")
    {
        s.pop();
        int a = stoi(s.top());
        s.pop();
        int b = stoi(s.top());
        s.pop();

        string ans = to_string(b * a);
```

```

        s.push(ans);
    }
    else if (s.top() == "/")
    {
        s.pop();
        int a = stoi(s.top());
        s.pop();
        int b = stoi(s.top());
        s.pop();

        string ans = to_string(b / a);
        s.push(ans);
    }
    else if (s.top() == "^")
    {
        s.pop();
        int a = stoi(s.top());
        s.pop();
        int b = stoi(s.top());
        s.pop();

        int p = pow(b, a);

        string ans = to_string(p);
        s.push(ans);
    }
    else if (s.top() == ")")
    {
        s.pop();
        cout << s.top() << endl;
        break;
    }
}
return 0;
}

```

Ans to the Question Number – 8
convert an Infix expression into its equivalent
Postfix expression.

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

#define vl vector<long long int>
#define vi vector<int>
#define pb push_back
#define all(x) (x).begin(), (x).end()

#define ll long long int
#define ld long double

#define fr(i,n) for (ll i=0;i<n;i++)
#define fr1(i,n) for (ll i=1;i<=n;i++)

#define endl '\n'

int precedence(char c)
{
    if(c == '+' || c == '-') return 1;
    else if(c == '*' || c == '/') return 2;
    else if(c == '^') return 3;
    else return -1;
}

string InfixTOPostfix(stack<char>s,string infix)
{
    string postfix;

    for(ll i=0;i<infix.size();i++)
    {
```

```

        if((infix[i] >= 'a' && infix[i]<='z') ||
(infix[i] >= 'A' && infix[i] <= 'Z'))
        {
            postfix = postfix + infix[i];
        }
    else if(infix[i] == '(')
    {
        s.push(infix[i]);
    }
    else if(infix[i] == ')')
    {
        while((s.top() != '(') &&
(!s.empty()))
        {
            char temp = s.top();
            postfix = postfix + temp;
            s.pop();
        }
        if(s.top() == '(')
        {
            s.pop();
        }
    }
    else if(infix[i] == '+' || infix[i] ==
'-' || infix[i] == '*' || infix[i] == '/' ||
infix[i] == '^')
    {
        if(s.empty())
        {
            s.push(infix[i]);
        }
        else if((precedence(infix[i]) ==
precedence(s.top())) && (infix[i] == '^'))
        {
            s.push(infix[i]);
        }
        else
        {

```

```

        while ((!s.empty()) &&
(precedence(infix[i]) <= precedence(s.top())))
        {
            char temp = s.top();
            postfix = postfix + temp;
            s.pop();
        }
        s.push(infix[i]);
    }
}

while(!s.empty())
{
    char temp = s.top();
    postfix = postfix + temp;
    s.pop();
}

return postfix;
}

int main()
{
    ///Peace be with you.

    string infix_exp,postfix_exp;
    cout << "Enter a Infix Expression : " <<
endl;
    cin >> infix_exp;

    cout << "Infix Expression : " << infix_exp
<< endl;

    stack<char> stack;
    postfix_exp =
InfixTOPostfix(stack,infix_exp);
    cout << endl;

```

```

        cout << "Postfix Expression : " <<
postfix_exp << endl;

    return 0;
}

/*
A+ (B*C- (D/E^F) *G) *H
*/

```

Ans to the Question Number - 9

(A)

Find the length of a string S

```

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

int main()
{
    string str;
    getline(cin, str);

    int cnt = 0;
    for (int i = 0; str[i] != '\0'; i++)
    {
        cnt++;
    }
    cout << "The Length of the string is: " <<
cnt;
    return 0;
}

```

(B)

Copy string S2 to S1.

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

int main()
{
    string str1, str2;
    getline(cin, str1);
    for (int i = 0; str1[i] != '\0'; i++)
    {
        str2 += str1[i];
    }

    cout << "The copied string is : " << str2;
    return 0;
}
```

(C)

Concatenate string S2 to S1.

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

int main()
{
    int i, j;
    char str1[200], str2[200];

    cout << "Enter 1st string: ";
    cin.getline(str1, 200);
    cout << "Enter 2nd string: ";
    cin.getline(str2, 200);
}
```



```

while (str1[i] != '\0')
{
    i++;
}
j = 0;
while (str2[j] != '\0')
{
    str1[i] = str2[j];
    i++;
    j++;
}
str1[i] = '\0';
cout << "After Concatenate, the string is : "
<< str1;
return 0;
}

```



Compare two strings S1 and S2

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i, j;

    string str1, str2;
    cout << "Enter 1st string: ";
    getline(cin, str1);
    cout << "Enter 2nd string: ";
    getline(cin, str2);

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

```

```

    {
        j++;
    }
    int temp = 1;
    if (i != j)
        temp = 0;
    else
    {
        for (i = 0, j = 0; str1[i] != '\0'; i++,
j++)
        {
            if (str1[i] != str2[j])
            {
                temp = 0;
                break;
            }
        }
    }

    if (temp == 1)
        cout << "Strings are equal" << endl;
    else
        cout << "Strings are not equal" << endl;
    return 0;
}

```

(E)

Reverse a string S.

```

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

int main()
{
    string str;
    cout<<"Enter the String: ";
    getline(cin,str);
}

```

```

int n=0;
while (str[n] != '\0')
{
    n++;
}

for (int i = 0; i < n / 2; i++)
{
    char ch = str[i];
    str[i] = str[n - i - 1];
    str[n - i - 1] = ch;
}
cout<<"String After Reverse: "<<str;

return 0;
}

```

Ans to the Question Number – 10
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()
{
    string str, add;
    int n;
    getline(cin, str);
    cin >> add >> n;

    int l1 = str.size();
    int l2 = add.size();
    int sz = l1 + l2;

```

```

    int i, j;
    for (i = sz - 1, j = 11 - 1; i > n; i--, j--)
    {
        str[i] = str[j];
    }

    for (i = n - 1, j = 0; j < 12; i++, j++)
    {
        str[i] = add[j];
    }
    for (i = 0; i < sz; i++)
    {
        cout << str[i];
    }
    return 0;
}

```

Ans to the Question Number – 11

**A text T in memory. delete a string S of length L
from Kth position in T.**

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    string str, newstr;
    getline(cin, str);

    int length, pos, i = 0;
    cin >> length;
    cin >> pos;
    int strLen = str.length();
    for (i = 0; i < pos; i++)
    {
        newstr[i] = str[i];
    }
}

```

```

        for (int j = (length + pos); j < strLen;
j++)
        {
            newstr[i] = str[j];
            i++;
        }
        cout << "Editted Text is : ";
        for (int k = 0; k < i; k++)
        {
            cout << newstr[k];
        }
        return 0;
    }
}

```

Ans to the Question Number – 12 **first occurrence of a pattern (P) in the string S.**

```

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

int main()
{
    string str1, str2;
    cin >> str1 >> str2;
    int maxx = str1.size() - str2.size() + 1;
    int ans;

    for (int i = 0; i < maxx; i++)
    {
        bool found = 0;

        for (int j = 0; j < str2.size(); j++)
        {

```

```

        if (str2[j] != str1[j + i])
        {
            found = 1;
        }
    }

    if (found == 0)
    {
        ans=i+1;
        break;
    }
}

if(ans>0)
{
    cout << "Found at index: " << ans <<
endl;
}
else
{
    cout<<"Not found"<<endl;
}

return 0;
}

```

Ans to the Question Number – 13
replace the first occurrence of a pattern (P) in T
by Q.

```

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

int main()
{

    string str1, str2, str3;

```

```

cin >> str1 >> str2 >> str3;
int len1 = str1.size();
int len2 = str2.size();
int len3 = str3.size();
int tolen = len1 - len2 + len3;
int l1 = len1 - len2 ;

int maxx = str1.size() - str2.size() + 1;
int pos;

for (int i = 0; i < maxx; i++)
{
    bool found = 0;

    for (int j = 0; j < str2.size(); j++)
    {
        if (str2[j] != str1[j + i])
        {
            found = 1;
        }
    }

    if (found == 0)
    {
        pos = i + 1;
        break;
    }
}

if (pos > 0)
{
    for (int i = pos - 1; i < len1; i++)
    {
        str1[i] = str1[i + len2];
    }

    string newstr;
    for (int i = 0; i < pos - 1; i++)

```

```

        {
            newstr[i] = str1[i];
        }
        int l = pos - 1;
        for (int i = 0; i < len3; i++)
        {
            newstr[l] = str3[i];
            l++;
        }

        for (int i = pos - 1; i < str1.size();
i++)
        {
            newstr[l] = str1[i];
            l++;
        }

        for (int i = 0; i < (str1.size() +
len3); i++)
        {
            cout << newstr[i];
        }

    }
    else
    {
        cout << "Not found" << endl;
    }

    return 0;
}

```


Ans to the Question Number – 14
calculates the no. of occurrence of each letter of
an input text.

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

int main()
{
    string s;
    getline(cin,s);
    int frq[26] = {0};

    int i, j;
    while (s[i] != '\0')
    {
        if (s[i] >= 'a' && s[i] <= 'z')
        {
            j = s[i] - 'a';
            frq[j]++;
        }
        i++;
    }

    for (i = 0; i < 26; i++)
    {
        if (frq[i] > 0)
        {
            cout << char(i + 'a') << "
occurrence " << frq[i] << " times." << endl;
        }
    }
    return 0;
}
```

Ans to the Question Number – 15

read a positive integer in base b ($2 \leq b \leq 16$) and convert it into base d ($2 \leq d \leq 16$).

```
#include<iostream>
using namespace std;
int main()
{
    int b, d, i, j , k, value, arr[200];
    string s;
    cin >> s;
    cin >> b >> d;
    int ls;
    ls = s.size();
    int m = 1, decimal = 0;
    for (i = ls - 1; i >= 0; i--)
    {
        if ('0' <= s[i] && s[i] <= '9')
            value = (int)s[i] - '0';
        else value = (int)s[i] - 'A' + 10;
        decimal += (value * m);
        m *= b;
    }
    j = 0;
    while (decimal)
    {
        arr[j++] = decimal % d;
        decimal /= d;
    }
    k = j;
    for (i = j - 1; i >= 0; i--)
    {
        if (arr[i] >= 10)
            cout << (char)(arr[i] - 10 + 'A');
        else
            cout << arr[i];
    }
    return 0;
}
```

```
}
```

Ans to the Question Number – 16
the Greatest Common Divisor (GCD) & Least
Common Multiple (LCM) of two given positive
integers.

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

int main()
{
    int x, y, a, b, t, gcd, lcm;
    cout << "Enter two integers: " << endl;
    cin >> x >> y;
    a = x;
    b = y;
    while (b != 0)
    {
        t = b;
        b = a % b;
        a = t;
    }
    gcd = a;
    lcm = (x / gcd) * y;

    cout << "GCD: " << gcd << endl;
    cout << "LCM: " << lcm << endl;
    return 0;
}
```