Q2) #include <iostream>

using namespace std;

int main() {

    int a[] = {64, 34, 25, 12, 22, 11, 90};

    int n = 7;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n - 1; j++) {

            if (a[j] > a[j + 1]) {

                int temp = a[j];

                a[j] = a[j + 1];

                a[j + 1] = temp;

            }

        }

    }

    for (int i = 0; i < n; i++)

        cout << a[i] << " ";

    return 0;

}

Q3) 1.) #include <iostream>

using namespace std;

int main() {

    int n;

    cout << "Enter the value of n : ";

    cin >> n;

    int arr[100];

    cout << "Enter " << n - 1 << " sorted elements from 1 to " << n << " (one missing): ";

    for (int i = 0; i < n - 1; i++)

        cin >> arr[i];

    for (int i = 0; i < n - 1; i++) {

        if (arr[i] != i + 1) {

            cout << "Missing number is: " << i + 1 << endl;

            return 0;

        }

    }

    cout << "Missing number is: " << n << endl;

    return 0;

}

Q3) 2.) #include <iostream>

using namespace std;

int Missingnumber(int arr[], int n) {

    int low = 0, high = n - 2;

    while (low <= high) {

        int mid = (low + high) / 2;

        if (arr[mid] == mid + 1)

            low = mid + 1;

        else

            high = mid - 1;

    }

    return low + 1;

}

int main() {

    int arr[] = {1, 2, 3, 5};

    int n = 5;

    cout << "Missing number (Binary Search): " << Missingnumber(arr, n);

    return 0;

}

Q4).1) #include <iostream>

#include <string>

using namespace std;

int main() {

    string str1, str2;

    cout << "Enter first string: ";

    cin >> str1;

    cout << "Enter second string: ";

    cin >> str2;

    string result = str1 + str2;

    cout << "Final string: " << result;

    return 0;

}

Q4).2) #include <iostream>

#include <string>

using namespace std;

int main() {

    string str;

    cout << "Enter a string: ";

    cin >> str;

    for (int i = str.length() - 1; i >= 0; i--)

        cout << str[i];

    return 0;

}

Q4).3) #include <iostream>

using namespace std;

int main() {

    string s, r;

    cin >> s;

    for (int i = 0; i < s.size(); i++)

        if (s[i] != 'a' && s[i] != 'e' && s[i] != 'i' &&

            s[i] != 'o' && s[i] != 'u')

            r += s[i];

    cout << r;

    return 0;

}

Q4).4) #include <iostream>

using namespace std;

int main() {

    string s[5];

    for (int i = 0; i < 5; i++) cin >> s[i];

    for (int i = 0; i < 4; i++)

        for (int j = 0; j < 4 - i; j++)

            if (s[j] > s[j + 1])

                swap(s[j], s[j + 1]);

    for (int i = 0; i < 5; i++) cout << s[i] << endl;

    return 0;

}

Q4).5) #include <iostream>

using namespace std;

int main() {

    char ch;

    cout << "Enter an uppercase character: ";

    cin >> ch;

    if (ch >= 'A' && ch <= 'Z')

        ch = ch + 32;

    cout << "Lowercase: " << ch;

    return 0;

}

Ques-5)

Diagonal elements-

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter size of diagonal matrix: ";

cin >> n;

int arr[n]; // Store only diagonal elements

cout << "Enter diagonal elements:\n";

for(int i = 0; i < n; i++) {

cin >> arr[i];

}

cout << "Matrix is:\n";

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(i == j) cout << arr[i] << " ";

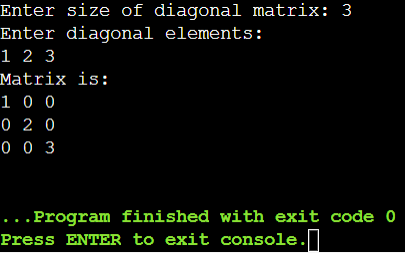
else cout << "0 ";

}

cout << endl;

}

}



b) Lower triangular matrix-

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter size of lower triangular matrix: ";

cin >> n;

int arr[n\*(n+1)/2];

cout << "Enter non-zero elements row-wise:\n";

for(int i = 0; i < n\*(n+1)/2; i++) cin >> arr[i];

int k = 0;

cout << "Matrix is:\n";

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(j <= i) cout << arr[k++] << " ";

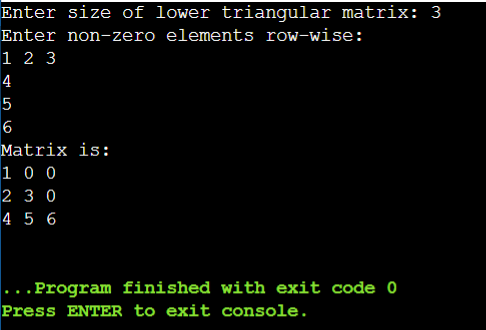
else cout << "0 ";

}

cout << endl;

}

}



Upper triangular matrix-

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter size of upper triangular matrix: ";

cin >> n;

int arr[n\*(n+1)/2];

cout << "Enter non-zero elements row-wise:\n";

for(int i = 0; i < n\*(n+1)/2; i++) cin >> arr[i];

int k = 0;

cout << "Matrix is:\n";

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(j >= i) cout << arr[k++] << " ";

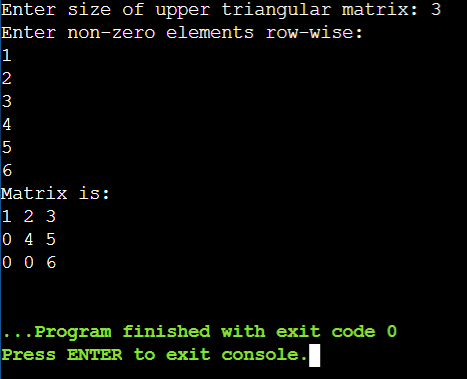
else cout << "0 ";

}

cout << endl;

}

}



Symmetrical matrix-

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter size of symmetric matrix: ";

cin >> n;

int arr[n\*(n+1)/2];

cout << "Enter lower triangle elements row-wise:\n";

for(int i = 0; i < n\*(n+1)/2; i++) cin >> arr[i];

cout << "Matrix is:\n";

int k = 0;

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(j <= i) cout << arr[k++] << " ";

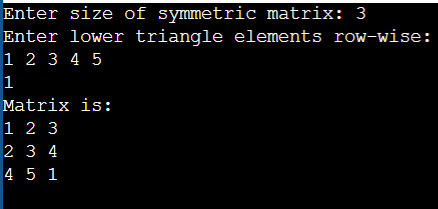
else cout << arr[i\*(i+1)/2 + j] << " "; // fetch symmetric

}

cout << endl;

}

}



Ques-6)

#include <iostream>

using namespace std;

int main() {

int A[100][3], T[100][3];

int rows, cols, nonZero;

int i, j, k = 0;

cout << "Enter rows, cols, and number of non-zero elements: ";

cin >> rows >> cols >> nonZero;

cout << "Enter triplets (row col value):\n";

for (i = 0; i < nonZero; i++)

cin >> A[i][0] >> A[i][1] >> A[i][2];

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

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

if (A[j][1] == i) {

T[k][0] = A[j][1];

T[k][1] = A[j][0];

T[k][2] = A[j][2];

k++;

}

}

}

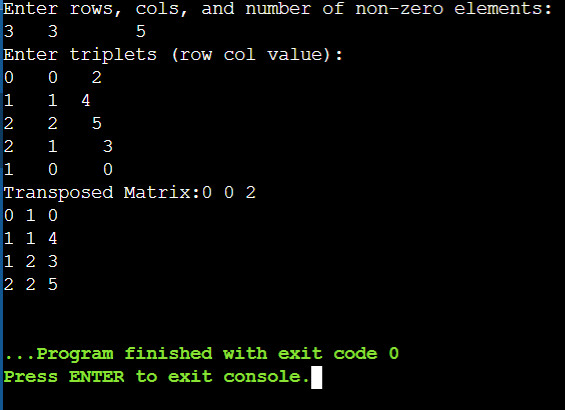
cout << "Transposed Matrix:";

for (i = 0; i < k; i++)

cout << T[i][0] << " " << T[i][1] << " " << T[i][2] << "\n";

return 0;

}



Addition of sparx matrix-

#include <iostream>

using namespace std;

int main() {

int A[100][3], B[100][3], SUM[200][3];

int m1, n1, t1, m2, n2, t2;

int i = 0, j = 0, k = 0;

cout << "Enter rows, cols, and non-zero count for Matrix A: ";

cin >> m1 >> n1 >> t1;

cout << "Enter triplets for Matrix A:\n";

for (int x = 0; x < t1; x++)

cin >> A[x][0] >> A[x][1] >> A[x][2];

cout << "Enter rows, cols, and non-zero count for Matrix B: ";

cin >> m2 >> n2 >> t2;

cout << "Enter triplets for Matrix B:\n";

for (int x = 0; x < t2; x++)

cin >> B[x][0] >> B[x][1] >> B[x][2];

if (m1 != m2 || n1 != n2) {

cout << "Matrix dimensions mismatch!\n";

return 0;

}

while (i < t1 && j < t2) {

if (A[i][0] == B[j][0] && A[i][1] == B[j][1]) {

SUM[k][0] = A[i][0];

SUM[k][1] = A[i][1];

SUM[k][2] = A[i][2] + B[j][2];

i++; j++; k++;

} else if (A[i][0] < B[j][0] || (A[i][0] == B[j][0] && A[i][1] < B[j][1])) {

SUM[k][0] = A[i][0];

SUM[k][1] = A[i][1];

SUM[k][2] = A[i][2];

i++; k++;

} else {

SUM[k][0] = B[j][0];

SUM[k][1] = B[j][1];

SUM[k][2] = B[j][2];

j++; k++;

}

}

while (i < t1) {

SUM[k][0] = A[i][0];

SUM[k][1] = A[i][1];

SUM[k][2] = A[i][2];

i++; k++;

}

while (j < t2) {

SUM[k][0] = B[j][0];

SUM[k][1] = B[j][1];

SUM[k][2] = B[j][2];

j++; k++;

}

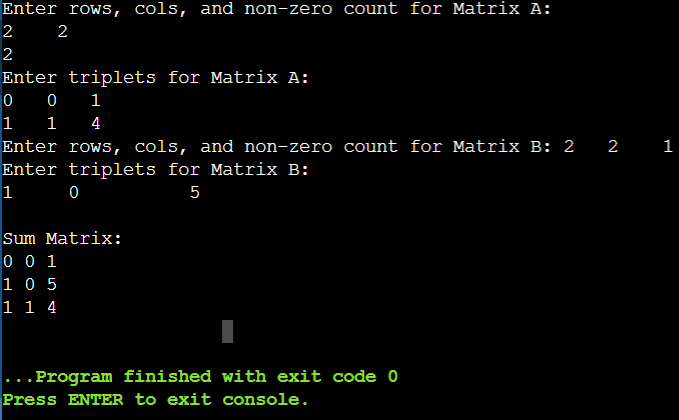
cout << "\nSum Matrix:\n";

for (i = 0; i < k; i++)

cout << SUM[i][0] << " " << SUM[i][1] << " " << SUM[i][2] << "\n";

return 0;

}



c)multiplication of sparx matrix-

#include <iostream>

using namespace std;

int main() {

int A[100][3], B[100][3], PROD[100][3];

int m1, n1, t1, m2, n2, t2;

int i, j, k, p = 0;

cout << "Enter rows, cols, and non-zero count for Matrix A: ";

cin >> m1 >> n1 >> t1;

cout << "Enter triplets for Matrix A:\n";

for (i = 0; i < t1; i++)

cin >> A[i][0] >> A[i][1] >> A[i][2];

cout << "Enter rows, cols, and non-zero count for Matrix B: ";

cin >> m2 >> n2 >> t2;

cout << "Enter triplets for Matrix B:\n";

for (i = 0; i < t2; i++)

cin >> B[i][0] >> B[i][1] >> B[i][2];

if (n1 != m2) {

cout << "Matrix dimensions mismatch!\n";

return 0;

}

for (i = 0; i < m1; i++) {

for (j = 0; j < n2; j++) {

int sum = 0;

for (k = 0; k < t1; k++) {

if (A[k][0] == i) {

for (int l = 0; l < t2; l++) {

if (B[l][0] == A[k][1] && B[l][1] == j)

sum += A[k][2] \* B[l][2];

}

}

}

if (sum != 0) {

PROD[p][0] = i;

PROD[p][1] = j;

PROD[p][2] = sum;

p++;

}

}

}

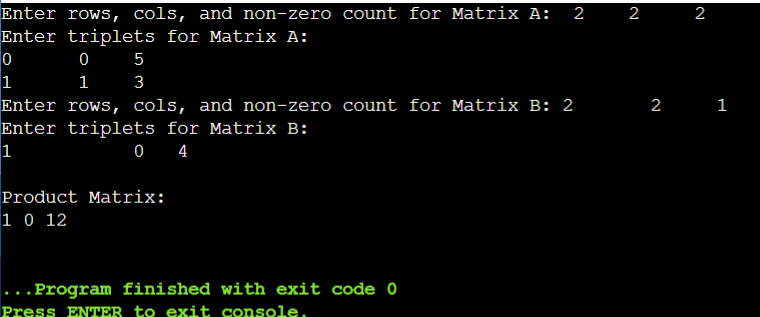
cout << "\nProduct Matrix:\n";

for (i = 0; i < p; i++)

cout << PROD[i][0] << " " << PROD[i][1] << " " << PROD[i][2] << "\n";

return 0;

}



Ques-7)

#include <iostream>

using namespace std;

int main() {

int A[100], n, i, j, count = 0;

cout << "Enter the number of elements in the array: ";

cin >> n;

cout << "Enter " << n << " real numbers:\n";

for (i = 0; i < n; i++)

cin >> A[i];

// Count inversions: i < j and A[i] > A[j]

for (i = 0; i < n - 1; i++) {

for (j = i + 1; j < n; j++) {

if (A[i] > A[j])

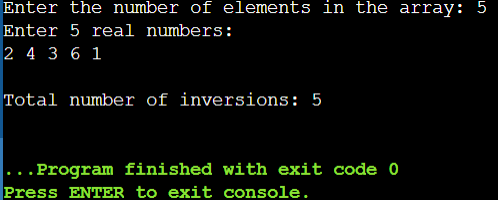
count++;

}

}

cout << "\nTotal number of inversions: " << count << "\n";

return 0;

}

Ques-8)

#include <iostream>

using namespace std;

int main() {

int A[100], n, i, j, count = 0;

bool isDistinct;

cout << "Enter number of elements: ";

cin >> n;

cout << "Enter " << n << " elements:\n";

for (i = 0; i < n; i++)

cin >> A[i];

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

isDistinct = true;

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

if (A[i] == A[j]) {

isDistinct = false;

break;

}

}

if (isDistinct)

count++;

}

cout << "\nTotal distinct elements: " << count << "\n";

return 0;

}