Q1. Implement the Binary search algorithm regarded as a fast search algorithm with run-time complexity of Ο(log n) in comparison to the Linear Search.

CODE :

#include <iostream>

using namespace std;

int main(){

    int num,ele,low,mid,high;

    cout<<"Please enter how many elements to be enetered into the array:    ";

    cin>>num;

    int arr[num];

    cout<<"Please enter the elements in a sorted manner\n";

    for(int i = 0;i<num;i++){

        cin>>arr[i];

    }

    low = 0;

    high = num-1;

    bool found = false;

    cout<<"Please eneter the element to be found:   ";

    cin>>ele;

    while(low<=high){

        mid = (low+high)/2;

        if(arr[mid]==ele){

            cout<<"The element was found at "<<mid+1<<" position";

            found = true;

            break;

        }

        else if(arr[mid]>ele){

            high = mid-1;

        }

        else{

            low = mid+1;

        }

    }

    if(found == false){

        cout<<"Element not present in the array";

    }

}

OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.

Q2. Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements:

CODE:

#include <iostream>

using namespace std;

int main(){

    int num,i,j,temp;

    cout<<"Please enter how many elements to be entered into the array:    ";

    cin>>num;

    int arr[num];

    cout<<"Please enter the elements in the array:  ";

    for(int i = 0;i<num;i++){

        cin>>arr[i];

    }

    for(i=0;i<num;i++){

        for(j=0;j<num-i-1;j++){

            if(arr[j]>arr[j+1]){

                temp = arr[j];

                arr[j] = arr[j+1];

                arr[j+1] = temp;

            }

        }

    }

    cout<<"Sorted array:    ";

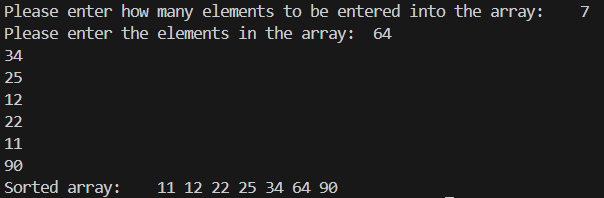
    for(int i = 0;i<num;i++){

        cout<<arr[i]<<" ";

    }

}

OUTPUT:



Q3. Design the Logic to Find a Missing Number in a Sorted Array.

CODE:

#include <iostream>

using namespace std;

int main(){

    int arr[4] = {2,3,5,6};

    for(int i =0;i<4;i++){

        if(arr[i+1] != arr[i]+1){

            cout<<"The missing number is:   "<<arr[i]+1;

            break;

        }

    }

}

OUTPUT:



Q4(a). Write a program to concatenate one string to another string.

CODE:

#include <iostream>

using namespace std;

int main(){

    string s1,s2;

    s1 = "Hello";

    s2 = "World";

    cout<<s1+s2;

}

OUTPUT:



Q4(b). Write a program to reverse a string.

CODE:

#include <iostream>

#include <string>

using namespace std;

int main(){

    string s1,s2;

    cout<<"Please enter a string:   ";

    getline(cin,s1);

    for(int i=s1.length()-1;i>=0;i--){

        s2 += s1[i];

    }

    cout<<s2;

}

OUTPUT:



Q4 (c).  Write a program to delete all the vowels from the string.

CODE:

#include <iostream>

#include <string>

using namespace std;

int main(){

    string s1,s2;

    cout<<"Please eneter a string:  ";

    getline(cin,s1);

    for(int i=0;i<s1.length();i++){

        if(s1[i]=='a'||s1[i]=='A'||s1[i]=='e'||s1[i]=='E'||s1[i]=='i'||s1[i]=='I'||s1[i]=='o'||s1[i]=='O'||s1[i]=='u'||s1[i]=='U'){

            continue;

        }

        else{

            s2+=s1[i];

        }

    }

    cout<<s2;

}

OUTPUT:



Q4 (d). Write a program to sort the strings in alphabetical order.

CODE:

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

int main(){

    string s1;

    cout<<"Please enter a string:   ";

    getline(cin,s1);

    sort(s1.begin(), s1.end());

    cout<<s1;

}

OUTPUT:



Q4 (e). Write a program to convert a character from uppercase to lowercase.

CODE:

#include <iostream>

#include <string>

using namespace std;

int main(){

    string s1;

    cout<<"Please enter a string:   ";

    getline(cin,s1);

    for(int i=0;i<s1.length();i++){

        if(65<=s1[i]&& s1[i]<=90){

            s1[i] += 32;

        }

    }

    cout<<s1;

}

OUTPUT:



Q5(a). Diagonal Matrix.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the size of the matrix:  ";

    cin>>num;

    int arr[num];

    cout<<"Please enter the diagonal elements:  ";

    for(int i=0;i<num;i++){

        cin>>arr[i];

    }

    cout<<"The 2D matrix is:    "<<"\n";

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i==j){

                cout<<arr[i]<<" ";

            }

            else{

                cout<<"0"<<" ";

            }

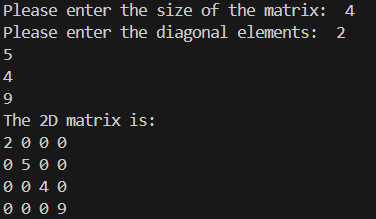
        }

        cout<<"\n";

    }

}

OUTPUT:



Q5.(b) Tri-diagonal Matrix.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the size of the matrix: ";

    cin>>num;

    int size = 3\*num-2;

    int arr[size];

    cout<<"Please enter the elements in the array:   ";

    for(int i=0;i<size;i++){

        cin>>arr[i];

    }

    cout<<"The 2D matrix is:    \n";

    int k = 0;

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i==j){

                cout<<arr[k]<<" ";

                k++;

            }

            else if(i-j==-1){

                cout<<arr[k]<<" ";

                k++;

            }

            else if(i-j==1){

                cout<<arr[k]<<" ";

                k++;

            }

            else{

                cout<<"0"<<" ";

            }

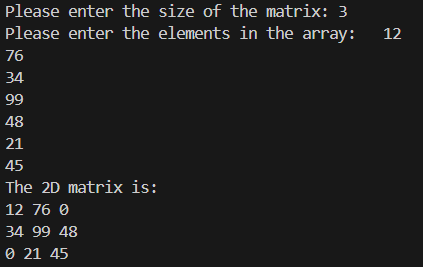
        }

        cout<<"\n";

    }

}

OUTPUT:



Q5. (c) Lower triangular Matrix.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the size of the matrix: ";

    cin>>num;;

    int size = ((num\*(num+1))/2);

    int arr[size];

    cout<<"Please enter the elements in the array:   ";

    for(int i=0;i<size;i++){

        cin>>arr[i];

    }

    cout<<"The 2D matrix is:    \n";

    int k=0;

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i>=j){

                cout<<arr[k]<<" ";

                k++;

            }

            else{

                cout<<"0"<<" ";

            }

        }

        cout<<"\n";

    }

}

OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.

Q5. (d) Upper triangular Matrix.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the size of the matrix: ";

    cin>>num;;

    int size = ((num\*(num+1))/2);

    int arr[size];

    cout<<"Please enter the elements in the array:   ";

    for(int i=0;i<size;i++){

        cin>>arr[i];

    }

    cout<<"The 2D matrix is:    \n";

    int k=0;

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i<=j){

                cout<<arr[k]<<" ";

                k++;

            }

            else{

                cout<<"0"<<" ";

            }

        }

        cout<<"\n";

    }

}

OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.

Q5. (e) Symmetric Matrix

CODE:

#include<iostream>

using namespace std;

int main(){

    int num;

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

    cin>>num;

    int size=(num\*(num+1))/2;

    int arr[size];

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

    for(int i=0;i<size;i++){

        cin>>arr[i];

    }

    cout<<"The 2D matrix is:    \n";

    int b[num][num];

    int k=0;

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i<=j){

                b[i][j]=arr[k];

                k++;

            }

        }

    }

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            if(i>j){

                b[i][j]=b[j][i];

            }

        }

    }

    for(int i=0;i<num;i++){

        for(int j=0;j<num;j++){

            cout<<b[i][j]<<" ";

        }

        cout<<"\n";

    }

}

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.

Q6. (a) Transpose of a matrix.

CODE:

#include <iostream>

using namespace std;

int main() {

int row, col;

int trans[100][3];

int s[100][3];

cout << "Enter the number of rows of array: ";

cin >> row;

cout << "Enter the number of columns of array: ";

cin >> col;

int arr[row][col];

cout << "Enter the elements of array:\n";

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

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

cin >> arr[i][j];

}

}

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

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

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

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

}

cout << "\n";

}

int k = 1;

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

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

if (arr[i][j] != 0) {

s[k][0] = i;

s[k][1] = j;

s[k][2] = arr[i][j];

k++;

}

}

}

s[0][0] = row;

s[0][1] = col;

s[0][2] = k - 1;

cout << "\nSparse Matrix (Triplet form):\n";

cout << "Row Column Value\n";

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

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

}

int x = 1;

for (int i = 0; i < s[0][1]; i++) {

for (int j = 1; j <= s[0][2]; j++) {

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

trans[x][0] = s[j][1];

trans[x][1] = s[j][0];

trans[x][2] = s[j][2];

x++;

}

}

}

trans[0][0] = s[0][1];

trans[0][1] = s[0][0];

trans[0][2] = s[0][2];

cout << "\nTranspose of Sparse Matrix:\n";

cout << "Row Column Value\n";

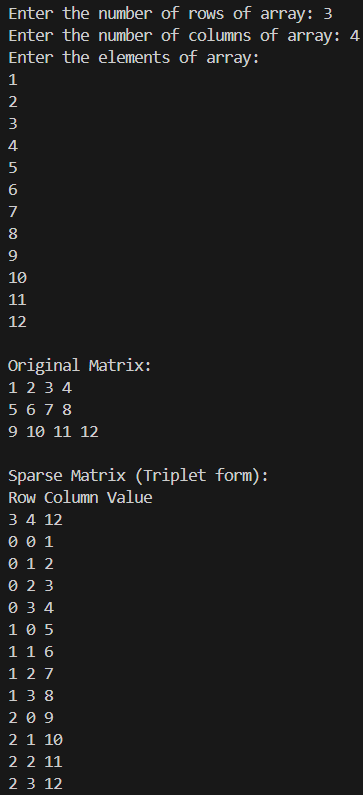
for (int i = 0; i < k; i++) {

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

}

}

OUTPUT:



A screenshot of a computer

AI-generated content may be incorrect.

Q6. (b)  Addition of two matrices.

CODE:

#include <iostream>

using namespace std;

int main() {

int row, col, arr1[100][100], arr2[100][100];

int s1[100][3], s2[100][3], add[200][3];

cout << "Enter the number of rows: ";

cin >> row;

cout << "Enter the number of columns: ";

cin >> col;

cout << "Enter elements of Matrix 1:\n";

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

cin >> arr1[i][j];

cout << "Enter elements of Matrix 2:\n";

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

cin >> arr2[i][j];

int k = 1, z = 1;

for (int i = 0; i < row; i++)

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

if (arr1[i][j] != 0) {

s1[k][0] = i;

s1[k][1] = j;

s1[k][2] = arr1[i][j];

k++;

}

if (arr2[i][j] != 0) {

s2[z][0] = i;

s2[z][1] = j;

s2[z][2] = arr2[i][j];

z++;

}

}

s1[0][0] = s2[0][0] = row;

s1[0][1] = s2[0][1] = col;

s1[0][2] = k - 1;

s2[0][2] = z - 1;

cout << "\nSparse Matrix 1 (Row Col Value):\n";

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

cout << s1[i][0] << " " << s1[i][1] << " " << s1[i][2] << "\n";

cout << "\nSparse Matrix 2 (Row Col Value):\n";

for (int i = 0; i < z; i++)

cout << s2[i][0] << " " << s2[i][1] << " " << s2[i][2] << "\n";

int m = 1, i = 1, j = 1;

add[0][0] = row;

add[0][1] = col;

while (i < k && j < z) {

if (s1[i][0] < s2[j][0] || (s1[i][0] == s2[j][0] && s1[i][1] < s2[j][1])) {

for (int p = 0; p < 3; p++)

add[m][p] = s1[i][p];

i++;

m++;

} else if (s2[j][0] < s1[i][0] || (s1[i][0] == s2[j][0] && s2[j][1] < s1[i][1])) {

for (int p = 0; p < 3; p++)

add[m][p] = s2[j][p];

j++;

m++;

} else {

int sum = s1[i][2] + s2[j][2];

if (sum != 0) {

add[m][0] = s1[i][0];

add[m][1] = s1[i][1];

add[m][2] = sum;

m++;

}

i++;

j++;

}

}

while (i < k) {

for (int p = 0; p < 3; p++)

add[m][p] = s1[i][p];

i++;

m++;

}

while (j < z) {

for (int p = 0; p < 3; p++)

add[m][p] = s2[j][p];

j++;

m++;

}

add[0][2] = m - 1;

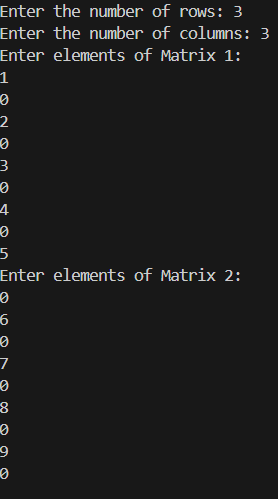
cout << "\nAddition Result (Sparse Matrix Form):\n";

for (int idx = 0; idx < m; idx++)

cout << add[idx][0] << " " << add[idx][1] << " " << add[idx][2] << "\n";

}

OUTPUT:



A screenshot of a computer program

AI-generated content may be incorrect.

Q6 (c)

CODE:

#include <iostream>

using namespace std;

int main() {

int r1, c1, r2, c2;

int arr1[100][100], arr2[100][100];

int s1[100][3], s2[100][3], trans[100][3], mult[100][3];

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 << "Multiplication not possible\n";

return 0;

}

cout << "Enter elements of matrix 1:\n";

for (int i = 0; i < r1; i++)

for (int j = 0; j < c1; j++)

cin >> arr1[i][j];

cout << "Enter elements of matrix 2:\n";

for (int i = 0; i < r2; i++)

for (int j = 0; j < c2; j++)

cin >> arr2[i][j];

int k = 1;

for (int i = 0; i < r1; i++)

for (int j = 0; j < c1; j++)

if (arr1[i][j] != 0) {

s1[k][0] = i;

s1[k][1] = j;

s1[k][2] = arr1[i][j];

k++;

}

s1[0][0] = r1;

s1[0][1] = c1;

s1[0][2] = k - 1;

int z = 1;

for (int i = 0; i < r2; i++)

for (int j = 0; j < c2; j++)

if (arr2[i][j] != 0) {

s2[z][0] = i;

s2[z][1] = j;

s2[z][2] = arr2[i][j];

z++;

}

s2[0][0] = r2;

s2[0][1] = c2;

s2[0][2] = z - 1;

int x = 1;

for (int col = 0; col < s2[0][1]; col++)

for (int i = 1; i <= s2[0][2]; i++)

if (s2[i][1] == col) {

trans[x][0] = s2[i][1];

trans[x][1] = s2[i][0];

trans[x][2] = s2[i][2];

x++;

}

trans[0][0] = s2[0][1];

trans[0][1] = s2[0][0];

trans[0][2] = s2[0][2];

int m = 1;

for (int i = 1; i <= s1[0][2]; i++) {

for (int j = 1; j <= trans[0][2]; j++) {

if (s1[i][1] == trans[j][1]) {

int r = s1[i][0];

int c = trans[j][0];

int val = s1[i][2] \* trans[j][2];

int found = 0;

for (int p = 1; p < m && !found; p++) {

if (mult[p][0] == r && mult[p][1] == c) {

mult[p][2] += val;

found = 1;

}

}

if (!found) {

mult[m][0] = r;

mult[m][1] = c;

mult[m][2] = val;

m++;

}

}

}

}

mult[0][0] = r1;

mult[0][1] = c2;

mult[0][2] = m - 1;

cout << "Resultant Sparse Matrix (Multiplication):\n";

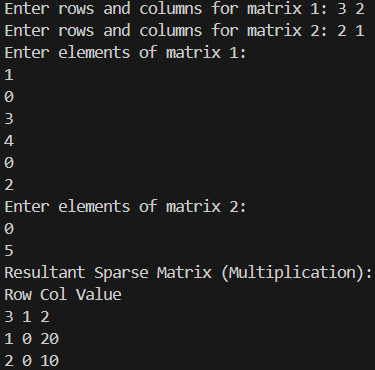
cout << "Row Col Value\n";

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

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

}

OUTPUT:



Q7. Let A[1 …. n] be an array of n real numbers. A pair (A[i], A[j ]) is said to be an inversion if these numbers are out of order, i.e., i < j but A[i]>A[j ]. Write a program to count the number of inversions in an array.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the number of elements in the array:    ";

    cin>>num;

    int arr[num];

    for(int i=0;i<num;i++){

        cin>>arr[i];

    }

    int countInversions = 0;

    for(int i=0;i<num;i++){

        for(int j=i+1;j<num;j++){

            if(arr[i]>arr[j]){

                countInversions+=1;

            }

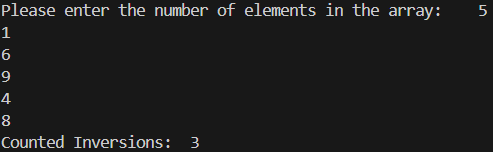
        }

    }

    cout<<"Counted Inversions:  "<<countInversions;

}

OUTPUT:



Q8. Write a program to count the total number of distinct elements in an array of length n.

CODE:

#include <iostream>

using namespace std;

int main(){

    int num;

    cout<<"Please enter the number of elements in the array:    ";

    cin>>num;

    int arr[num];

    for(int i=0;i<num;i++){

        cin>>arr[i];

    }

    int countUnique = 0;

    for(int i=0;i<num;i++){

        bool unique = true;

        for(int j=0;j<i;j++){

            if(arr[i]==arr[j]){

                unique = false;

                break;

            }

        }

        if(unique == true){

            countUnique += 1;

        }

    }

    cout<<"Number of unique elements:   "<<countUnique;

}

OUTPUT:

A black screen with white text

AI-generated content may be incorrect.