**DSA lab assignment 2**

**Q1.**

#include <iostream>

using namespace std;

int main() {

int n=7;

int arr[n]={11,12,13,14,15,17,18};

int start=0;

int end=n-1;

int x=15;

while(start<=end){

int mid=(start+end)/2;

if (arr[mid]<x){

start=mid+1;

}

else if(arr[mid]>x){

end=mid-1;

}

else if(arr[mid]==x){

cout<<"index of "<<x<<" is "<<mid<<endl;

break;

}

}

}

**Q2.**

#include <iostream>

using namespace std;

int main() {

int n=7;

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

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

bool flag=false;

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

if(arr[j]>arr[j+1]){

flag=true;

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

if(flag==false){

break;

}

}

cout<<"Sorted array is:-"<<endl;

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

cout<<arr[i]<<endl;

}

return 0;

}

**Q3.**

1. #include <iostream>

using namespace std;

int main() {

int n=7;

int arr[n]={12,13,15,16,17,18,19};

int start=0;

int end=n-1;

int x=15;

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

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

cout<<"missing element is "<<arr[0]+i;

break;

}

}

}

1. #include <iostream>

using namespace std;

int main(){

int n=6;

int arr[n]={11,12,13,14,15,17};

int start=0;

int end=n-1;

while(start<=end){

int mid=(start+end)/2;

if (arr[mid]==arr[0]+mid){

start=mid+1;

}

if(arr[mid]!=arr[0]+mid){

end=mid-1;

}

}

cout<<"missing element is "<<arr[0]+start;

}

**Q4.**

1. #include <iostream>

#include <cstring>

using namespace std;

int main() {

char a[10]="hello";

char b[10]=" world";

string c="HELLO ";

string d="WORLD";

strcat(a,b);

cout<<a<<endl;

cout<<c+d<<endl;

return 0;

}

1. #include <iostream>

#include <cstring>

using namespace std;

int main() {

string c="S O H A M";

int n=c.length();

for(int i=0;i<n/2;i++){

char temp;

temp=c[i];

c[i]=c[n-i-1];

c[n-i-1]=temp;

}

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

cout<<c[i];

}

return 0;

}

1. #include <iostream>

#include <cstring>

using namespace std;

bool vowel(char a){

if (a=='A'||a=='E'||a=='I'||a=='O'||a=='U'){

return true;

}

else if(a=='a'||a=='e'||a=='i'||a=='o'||a=='u'){

return true;

}

else{

return false;

}

}

int main() {

char j;

int count=0;

char b[50]="My name is Soham";

for(int i=0;i<strlen(b);i++){

if(vowel(b[i])==false){

b[count]=b[i];

count++;

}

}

b[count]='\0';

for (char j:b){

if(j=='\0'){

break;

}

cout<<j;

}

return 0;

}

1. #include <iostream>

#include <cstring>

using namespace std;

bool comp(char a,char b){

char lowA=tolower(a);

char lowB=tolower(b);

if(lowA==lowB){

if(a>b){

return true ;

}

else {

return false;

}

}

else if(lowA>lowB){

return true;

}

return false;

}

int main() {

int a='a';

char j;

int count=0;

string c="bacABcC";

for(int i=0;i<c.length();i++){

for(int j=0;j<c.length()-1;j++){

if(comp(c[j],c[j+1])==true){

char d=c[j];

c[j]=c[j+1];

c[j+1]=d;

}

}

}

for(int i=0;i<c.length();i++){

cout<<c[i];

}

return 0;}

1. #include <iostream>

using namespace std;

int main() {

char a='B';

if(a==tolower(a)){

cout<<a<<" is already in lower case"<<endl;

}

else{

cout<<" Lowercase of "<<a<<" is ";

a=tolower(a);

cout<<a;

}

return 0;}

**Q5.**

1. #include <bits/stdc++.h>

using namespace std;

int main() {

int n = 4;

int diag[n] = {1, 2, 3, 4};

cout << "Diagonal Matrix:\n";

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

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

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

else cout << 0 << " ";

}

cout << endl;

}

return 0;

}

b) #include <bits/stdc++.h>

using namespace std;

int main() {

int n = 4;

int tri[3\*n - 2] = {1,2,3,4, 5,6,7, 8,9,10};

cout << "Tri-diagonal Matrix:\n";

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

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

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

else if (i == j-1) cout << tri[n+i] << " ";

else if (i == j+1) cout << tri[2\*n-1+j] << " ";

else cout << 0 << " ";

}

cout << endl;

}

return 0;

}

c) #include <bits/stdc++.h>

using namespace std;

int main() {

int n = 4;

int lower[n\*(n+1)/2] = {1,2,3,4, 5,6,7, 8,9, 10};

cout << "Lower Triangular Matrix:\n";

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

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

if (i >= j) cout << lower[i\*(i+1)/2 + j] << " ";

else cout << 0 << " ";

}

cout << endl;

}

return 0;

}

d) #include <bits/stdc++.h>

using namespace std;

int main() {

int n = 4;

int upper[n\*(n+1)/2] = {1,2,3,4, 5,6,7, 8,9, 10};

cout << "Upper Triangular Matrix:\n";

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

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

if (j >= i) cout << upper[j\*(j+1)/2 + i] << " ";

else cout << 0 << " ";

}

cout << endl;

}

return 0;

}

e) #include <bits/stdc++.h>

using namespace std;

int main() {

int n = 4;

int sym[n\*(n+1)/2] = {1,2,3,4, 5,6,7, 8,9, 10};

cout << "Symmetric Matrix:\n";

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

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

if (i >= j) cout << sym[i\*(i+1)/2 + j] << " ";

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

}

cout << endl;

}

return 0;

}

**Q6.**

a) #include <bits/stdc++.h>

using namespace std;

struct Element {

int row, col, val;

};

int main() {

vector<Element> A = {{0,0,5}, {0,2,8}, {2,1,6}};

vector<Element> T;

for (auto &e : A) T.push\_back({e.col, e.row, e.val});

for (auto &e : T) cout << "("<<e.row<<","<<e.col<<","<<e.val<<") ";

cout << endl;

return 0;

}

b) #include <bits/stdc++.h>

using namespace std;

struct Element {

int row, col, val;

};

int main() {

vector<Element> A = {{0,0,5}, {0,2,8}, {2,1,6}};

vector<Element> B = {{0,1,7}, {2,1,6}};

vector<Element> C = A;

for (auto &b : B) C.push\_back(b);

for (auto &e : C) cout << "("<<e.row<<","<<e.col<<","<<e.val<<") ";

cout << endl;

return 0;

}

c) #include <bits/stdc++.h>

using namespace std;

struct Element {

int row, col, val;

};

int main() {

vector<Element> A = {{0,0,5}, {0,2,8}, {2,1,6}};

vector<Element> B = {{0,1,7}, {2,1,6}};

vector<Element> M;

for (auto &a : A) {

for (auto &b : B) {

if (a.col == b.row) {

M.push\_back({a.row, b.col, a.val \* b.val});

}

}

}

for (auto &e : M) cout << "("<<e.row<<","<<e.col<<","<<e.val<<") ";

cout << endl;

return 0;

}

**Q7.**

#include <bits/stdc++.h>

using namespace std;

int main() {

vector<int> arr = {2, 4, 1, 3, 5};

int n = arr.size();

int inv = 0;

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

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

if (arr[i] > arr[j]) inv++;

}

}

cout << "Inversions = " << inv << endl;

return 0;

}

**Q8.**

#include <bits/stdc++.h>

using namespace std;

int main() {

vector<int> arr = {1, 2, 2, 3, 4, 4, 5};

sort(arr.begin(), arr.end());

int cnt = 1;

for (int i=1; i<arr.size(); i++) {

if (arr[i] != arr[i-1]) cnt++;

}

cout << "Distinct elements = " << cnt << endl;

return 0;

}