SetA

#include <iostream>

using namespace std;

int main()

{ int count=0;

int arr[6]={1,2,3,-9,7,-8};

for(int i=0;i<6;i++)

{

if(arr[i]<0)

{

count++;

}

}

cout<<count;

}

#include <iostream>

using namespace std;

int main() {

int arr[] = { 2, 2, 0, -9, 6, 7 };

int n = sizeof(arr) / sizeof(int);

int max = arr[0]; // Initialize min with the first element

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

if (arr[i] > max) {

max = arr[i]; // Update min if current element is smaller

}

}

cout << "Maximum element: " << max<< endl;

return 0;

}

//Merge of 2 sorted array

#include<bits/stdc++.h>

using namespace std;

int main()

{

int n;

cout<<"Enter size of Arrays :";

cin>>n;

int a[n],b[n];

int m = n+n;

int c[m];

cout<<"Enter elements in array 1 :" ;

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

{

cin>>a[i];

}

cout<<"Enter elements in array 2 :" ;

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

{

cin>>b[i];

}

int i=0,j=0;

int k=0;

while(i<n || j<n)

{

if(a[i]<b[j])

{

c[k]=a[i];

i++,k++;

}

else

{

c[k]=b[j];

j++,k++;

}

}

cout<<"Final Sorted ";

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

{

cout<<c[k]<<" ";

}

return 0;

}

// Reverse Array

#include<bits/stdc++.h>

using namespace std;

int main()

{

cout<<"Enter 7 Integers in the Array";

int a[7];

for(int i=0;i<7;i++)

{

cin>>a[i];

}

int i=0,j=6,temp;

while(i < j)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

i++,j--;

}

for(i=0;i<7;i++)

{

cout<<a[i]<<" ";

}

return 0;

}

// Array is Palindrome or Not

#include<bits/stdc++.h>

using namespace std;

int main()

{

cout<<"Enter 5 Integers in the Array :";

int a[5],b[5];

for(int i=0;i<5;i++)

{

cin>>a[i];

}

int i=0,j=4,temp;

int count=0;

while(i < j)

{

if(a[i]==a[j])

{

count++;

}

i++,j--;

}

if(count==2) cout<<"Array is Palindrome";

else cout<<"Array is not Palindrome";

return 0;

}

#include <iostream>

using namespace std;

int main() {

int n, target, index = -1;

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

cin >> n;

int arr[n];

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

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

cin >> arr[i];

}

cout << "Enter the number to search: ";

cin >> target;

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

if (arr[i] == target) {

index = i;

break;

}

}

cout << "Index: " << index << endl; // Prints -1 if not found

return 0;

}

#include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

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

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

cin >> arr[i];

}

cout << "Unique numbers: ";

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

bool isUnique = true;

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

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

isUnique = false;

break;

}

}

if (isUnique) {

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

}

}

cout << endl;

return 0;

}

#include <iostream>

#include <map>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

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

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

cin >> arr[i];

}

map<int, int> frequency;

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

frequency[arr[i]]++;

}

cout << "Frequency of each number:" << endl;

for (auto it : frequency) {

cout << it.first << ": " << it.second << endl;

}

return 0;

}

#include <iostream>

#include <map>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

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

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

cin >> arr[i];

}

map<int, int> frequency;

int duplicateCount = 0;

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

frequency[arr[i]]++;

}

for (auto it : frequency) {

if (it.second > 1) {

duplicateCount++;

}

}

cout << "Total number of duplicate numbers: " << duplicateCount << endl;

return 0;

}

#include <stdio.h>

void printParallelogram(int rows, int cols) {

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

// Print leading spaces

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

printf(" ");

}

// Print stars

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

printf("\*");

}

printf("\n");

}

}

int main() {

printParallelogram(5, 5);

return 0;

}

#include <stdio.h>

void printRightTriangle(int rows) {

for (int i = 1; i <= rows; i++) {

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

printf("\*");

}

printf("\n");

}

}#include <stdio.h>

void printHalfDiamond(int cols) {

for (int i = 1; i <= cols; i++) {

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

printf("\*");

}

printf("\n");

}

for (int i = cols - 1; i >= 1; i--) {

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

printf("\*");

}

printf("\n");

}

}

int main() {

printHalfDiamond(5);

return 0;

}

#include <stdio.h>

int printDiamond(int rows) {

if (rows % 2 == 0) {

printf("Error: Number of rows must be odd.\n");

return 0;

}

int mid = rows / 2;

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

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

printf(" ");

}

for (int k = 0; k < 2 \* i + 1; k++) {

printf("\*");

}

printf("\n");

}

for (int i = mid - 1; i >= 0; i--) {

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

printf(" ");

}

for (int k = 0; k < 2 \* i + 1; k++) {

printf("\*");

}

printf("\n");

}

return 1;

}

int main() {

printDiamond(9);

return 0;

}

#include <stdio.h>

void print12345Pattern(int rows, int cols) {

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

if (i % 2 == 0) {

for (int j = 1; j <= cols; j++) {

printf("%d", j);

}

} else {

for (int j = cols; j >= 1; j--) {

printf("%d", j);

}

}

printf("\n");

}

}#include <stdio.h>

void printBorderPattern(int rows, int cols) {

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

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

if (i == 0 || i == rows - 1 || j == 0 || j == cols - 1) {

printf("1");

} else {

printf("0");

}

}

printf("\n");

}

}

int main() {

printBorderPattern(5, 5);

return 0;

}

#include <stdio.h>

void printDiagonalNumbers(int rows) {

for (int i = 1; i <= rows; i++) {

for (int j = i; j <= rows; j++) {

printf("%d", j);

}

for (int k = 1; k < i; k++) {

printf("%d", rows);

}

printf("\n");

}

}

int main() {

printDiagonalNumbers(5);

return 0;

}

int main() {

print12345Pattern(5, 5);

return 0;

}

int main() {

printRightTriangle(5);

return 0;

}