30.narcissistic number

#include <iostream>

#include <cmath> // for pow function

using namespace std;

bool isNarcissistic(int n) {

int original = n, numDigits = 0, sum = 0;

// Count the number of digits

while (original > 0) {

original /= 10;

numDigits++;

}

original = n;

// Calculate the sum of nth powers of digits

while (original > 0) {

int digit = original % 10;

sum += pow(digit, numDigits);

original /= 10;

}

return sum == n;

}

int main() {

int num;

cout << "Enter a number: ";

cin >> num;

if (isNarcissistic(num)) {

cout << num << " is a Narcissistic number!" << endl;

} else {

cout << num << " is not a Narcissistic number." << endl;

}

return 0;

}

31. print the pattern 1 22 333 4444 55555

#include <iostream>

#include <string>

using namespace std;

int main() {

for (int num = 1; num <= 5; num++) {

string repeatedNumber = string(num, char(num + '0'));

cout << repeatedNumber << endl;

}

return 0;

}

32. print the pattern \* \*\* \*\*\* \*\*\*\* \*\*\*\*\*

#include <iostream>

int main() {

int rows = 5; // Number of rows in the pattern

// Outer loop for the number of rows

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

// Inner loop for printing '\*' characters

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

std::cout << "\*";

}

std::cout << std::endl; // Move to the next line after each row

}

return 0;

}

33. Print pascal triangle pattern nested for loop

#include <iostream>

using namespace std;

int main() {

int rows;

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

cin >> rows;

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

// Print leading spaces for proper formatting

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

cout << " ";

}

// Calculate and print Pascal's coefficients

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

int coefficient = 1;

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

coefficient \*= (i - k);

coefficient /= (k + 1);

}

cout << coefficient << " ";

}

cout << endl;

}

return 0;

}

34. Print diamond pattern with \* using nested for loop

#include <iostream>

using namespace std;

int main() {

int rows;

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

cin >> rows;

// Print upper half

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

// Print leading spaces

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

cout << " ";

}

// Print stars (2 \* i - 1) times

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

// Print lower half (reverse order)

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

// Print leading spaces

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

cout << " ";

}

// Print stars (2 \* i - 1) times

for (int j = 1; j <= 2 \* i - 1; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

35. Program to reverse the elements in an array

#include <iostream>

using namespace std;

void reverseArray(int arr[], int size) {

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

swap(arr[i], arr[size - i - 1]);

}

}

void printArray(int arr[], int size) {

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

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

}

cout << endl;

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Original array: ";

printArray(arr, size);

reverseArray(arr, size);

cout << "Reversed array: ";

printArray(arr, size);

return 0;

}

36. Program to insert an element in an array at a specific position

#include <iostream>

using namespace std;

int main() {

int arr[100], pos, tot, elem;

cout << "Enter the Size for Array: ";

cin >> tot;

cout << "Enter " << tot << " Array Elements: ";

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

cin >> arr[i];

cout << "Enter Element to Insert: ";

cin >> elem;

cout << "At What Position? ";

cin >> pos;

for (int i = tot; i >= pos; i--)

arr[i] = arr[i - 1];

arr[pos] = elem;

tot++;

cout << "\nThe New Array is:\n";

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

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

cout << endl;

return 0;

}

37. Program to Delete an element in an array at a specific position

#include<iostream>

using namespace std;

int main()

{

int arr[10], tot=10, i, elem, j, found=0;

cout<<"Enter 10 Array Elements: ";

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

cin>>arr[i];

cout<<"\nEnter Element to Delete: ";

cin>>elem;

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

{

if(arr[i]==elem)

{

for(j=i; j<(tot-1); j++)

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

found++;

i--;

tot--;

}

}

if(found==0)

cout<<"\nElement doesn't found in the Array!";

else

cout<<"\nElement Deleted Successfully!";

cout<<endl;

return 0;

}

39. Find the average of all elements in an array

#include <iostream>

using namespace std;

// Function that return average

// of an array.

double average(int a[], int n)

{

// Find sum of array element

int sum = 0;

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

sum += a[i];

return (double)sum / n;

}

// Driver code

int main()

{

int arr[] = { 10, 2, 3, 4, 5, 6, 7, 8, 9 };

int n = sizeof(arr) / sizeof(arr[0]);

cout << average(arr, n) << endl;

return 0;

}

40. Find the second largest element in an array

#include <climits>

#include <iostream>

**using** **namespace** std;

**int** main()

{

    // Initialize an array

**int** array[] = { 1, 2, 3, 4, 5 };

**int** n = **sizeof**(array) / **sizeof**(array[0]);

    // Initialize first and second to the minimum possible

    // value

**int** first = INT\_MIN, second = INT\_MIN;

    // Traverse the array

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

        // If current element is greater than first

**if** (array[i] > first) {

            second = first;

            first = array[i];

        }

        // If current element is in between first and second

**else** **if** (array[i] > second && array[i] < first) {

            second = array[i];

        }

    }

    // Print the second largest element

    cout << "Second Largest Element in the Array: "

         << second << endl;

**return** 0;

}

41. Find the number of occurrences of a value in an array

#include <stdio.h>

// Returns number of times x occurs in arr[0..n-1]

int countOccurrences(int arr[], int n, int x)

{

int res = 0;

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

if (x == arr[i])

res++;

return res;

}

// Driver code

int main()

{

int arr[] = {1, 2, 2, 2, 2, 3, 4, 7, 8, 8};

int n = sizeof(arr) / sizeof(arr[0]);

int x = 2;

printf("%d", countOccurrences(arr, n, x));

return 0;

}

42. Merge two array

#include<bits/stdc++.h>

using namespace std;

void mergeArrays(int a[], int b[], int n, int m)

{

map<int, int> mp;

for(int i = 0; i < n; i++)mp[a[i]]++;

for(int i = 0;i < m;i++)mp[b[i]]++;

for(auto j: mp)

{

for(int i=0; i<j.second;i++)cout<<j.first<<" ";

}

}

int main()

{

int a[] = {1, 3, 5, 7}, b[] = {2, 4, 6, 8};

int size = sizeof(a)/sizeof(int);

int size1 = sizeof(b)/sizeof(int);

mergeArrays(a, b, size, size1);

return 0;

}

43. Create a dynamic array using pointers and display the values

#include <iostream>

int main() {

int size;

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

std::cin >> size;

int \*arr = new int[size];

std::cout << "Enter " << size << " elements:" << std::endl;

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

std::cin >> arr[i];

}

std::cout << "Values in the array:" << std::endl;

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

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

}

std::cout << std::endl;

delete[] arr;

return 0;

}

44.Create a dynamic 2D (Two dimensional) array using pointers and display the values

#include <iostream>

**using** **namespace** std;

**int** main()

{

**int** m = 3, n = 4, c = 0;

**int**\* arr = **new** **int**[m \* n];

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

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

            \*(arr + i \* n + j) = ++c;

        }

    }

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

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

            cout << \*(arr + i \* n + j)

                 << " ";

        }

        cout << endl;

    }

**delete**[] arr;

**return** 0;

}

45.Add 2 matrices

#include <iostream>

using namespace std;

int main()

{

int a[10][10],b[10][10], c[10][10] ,row, col,i,j;

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

cin>>row;

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

cin>>col;

cout<<"enter A matrix element : ";

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

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

cin>>a[i][j];

}

}

cout<<"enter B matrix element : ";

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

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

cin>>b[i][j];

}

}

cout<<"ADDITION OF MATRIX A&B : \n";

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

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

c[i][j]=a[i][j]+b[i][j];

cout<<c[i][j];

cout<<" ";

}

cout<<"\n";

}

   return 0;

}

46.Multiply 2 matrices

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

**using** **namespace** **std**;

*// Edit MACROs here, according to your Matrix Dimensions for*

*// mat1[R1][C1] and mat2[R2][C2]*

#define R1 2 *// number of rows in Matrix-1*

#define C1 2 *// number of columns in Matrix-1*

#define R2 2 *// number of rows in Matrix-2*

#define C2 2 *// number of columns in Matrix-2*

void mulMat(int mat1[][C1], int mat2[][C2])

{

int rslt[R1][C2];

cout << "Multiplication of given two matrices is:**\n**";

**for** (int i = 0; i < R1; i++) {

**for** (int j = 0; j < C2; j++) {

rslt[i][j] = 0;

**for** (int k = 0; k < R2; k++) {

rslt[i][j] += mat1[i][k] \* mat2[k][j];

}

cout << rslt[i][j] << "**\t**";

}

cout << endl;

}

}

*// Driver code*

int main()

{

*// R1 = 4, C1 = 4 and R2 = 4, C2 = 4 (Update these*

*// values in MACROs)*

int mat1[R1][C1] = { { 1, 1 },

{ 2, 2 } };

int mat2[R2][C2] = { { 1, 1 },

{ 2, 2 } };

**if** (C1 != R2) {

cout << "The number of columns in Matrix-1 must "

"be equal to the number of rows in "

"Matrix-2"

<< endl;

cout << "Please update MACROs according to your "

"array dimension in #define section"

<< endl;

exit(EXIT\_FAILURE);

}

*// Function call*

mulMat(mat1, mat2);

**return** 0;

}

47.Find the sum of diagonals of a matrix

#include <bits/stdc++.h>

**using** **namespace** std;

**const** **int** MAX = 100;

**void** printDiagonalSums(**int** mat[][MAX], **int** n)

{

**int** principal = 0, secondary = 0;

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

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

            // Condition for principal diagonal

**if** (i == j)

                principal += mat[i][j];

            // Condition for secondary diagonal

**if** ((i + j) == (n - 1))

                secondary += mat[i][j];

        }

    }

    cout << "Principal Diagonal:" << principal << endl;

    cout << "Secondary Diagonal:" << secondary << endl;

}

// Driver code

**int** main()

{

**int** a[][MAX] = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 },

                    { 1, 2, 3, 4 }, { 5, 6, 7, 8 } };

    printDiagonalSums(a, 4);

**return** 0;

}

Factorial in c++

1)FACTORIAL USING FUNCTION

#include <iostream>

using namespace std;

int main()

{

int i,fact=1,number;

cout<<"Enter any Number: ";

cin>>number;

for(i=1;i<=number;i++){

fact=fact\*i;

}

cout<<"Factorial of " <<number<<" is: "<<fact<<endl;

return 0;

}

2)PRIME NUMBER USING FUNCTION

#include <iostream>

using namespace std;

bool check\_prime(int);

int main() {

int n;

cout << "Enter a positive integer: ";

cin >> n;

if (check\_prime(n))

cout << n << " is a prime number.";

else

cout << n << " is not a prime number.";

return 0;

}

bool check\_prime(int n) {

bool is\_prime = true;

// 0 and 1 are not prime numbers

if (n == 0 || n == 1) {

is\_prime = false;

}

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

if (n % i == 0) {

is\_prime = false;

break;

}

}

return is\_prime;

}

3)REVERSE OF A STRING USING FUNCTION

#include <bits/stdc++.h>

using namespace std;

void reverseStr(string& str)

{

int len = str.length();

int n = len-1;

int i = 0;

while(i<=n){

swap(str[i],str[n]);

n = n-1;

i = i+1;

}

}

int main()

{

string str = "geeksforgeeks";

reverseStr(str);

cout << str;

return 0;

}

4)MIN AND MAX ELEMENTS USING ARRAY

#include <bits/stdc++.h>

using namespace std;

int main()

{

int arr[] = { 1, 45, 54, 71, 76, 12 };

int n = sizeof(arr) / sizeof(arr[0]);

cout << "Array: ";

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

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

cout << "\nMin Element = "

<< \*min\_element(arr, arr + n);

cout << "\nMax Element = "

<< \*max\_element(arr, arr + n);

int &min = \*min\_element(arr,arr+n );

int &max = \*max\_element(arr,arr+n );

cout<<"\nFinding the Element using address variable";

cout<<"\nMin Element = "<<min;

cout<<"\nMax Element = "<<max;

return 0;

}

5)GCD OF TWO NUMBERS USING FUNCTION

#include <iostream>

using namespace std;

int gcd(int a, int b) {

if (b == 0)

return a;

return gcd(b, a % b);

}

int main() {

int a = 105, b = 30;

cout<<"GCD of "<< a <<" and "<< b <<" is "<< gcd(a, b);

return 0;

}

6)FUNCTIONS TO COUNT ELEMENTS IN THE STRING

#include <bits/stdc++.h>

using namespace std;

int main()

{

int arr[] = { 3, 2, 1, 3, 3, 5, 3 };

int n = sizeof(arr) / sizeof(arr[0]);

cout <<

" Number of times 3 appears : "

<< count(arr, arr + n, 3);

return 0;

7)CONVERT CELCUIS TO FARHENHIT

#include <iostream>

using namespace std;

double celsiusToFahrenheit(double celsius) {

return (celsius \* 9 / 5) + 32;

}

double fahrenheitToCelsius(double fahrenheit) {

return (fahrenheit - 32) \* 5 / 9;

}

int main() {

double celsius, fahrenheit;

cout << "Enter temperature in Celsius: ";

cin >> celsius;

fahrenheit = celsiusToFahrenheit(celsius);

cout << "Temperature in Fahrenheit: " << fahrenheit << endl;

cout << "Enter temperature in Fahrenheit: ";

cin >> fahrenheit;

celsius = fahrenheitToCelsius(fahrenheit);

cout << "Temperature in Celsius: " << celsius << endl;

return 0;

}

8)AREA OF CIRCLE USING FUNCTION

#include <iostream>

using namespace std;

const double PI = 3.14159;

double calculateArea(double radius) {

return PI \* radius \* radius;

}

int main() {

double radius;

cout << "Enter the radius of the circle: ";

cin >> radius;

if (radius < 0) {

cout << "Radius cannot be negative." << endl;

return 1;

}

double area = calculateArea(radius);

cout << "The area of the circle with radius " << radius << " is: " << area << endl;

return 0;

}

9)PALINDROME USING FUNCTION

#include <iostream>

#include <string>

using namespace std;

bool isPalindrome(const string& str) {

int left = 0;

int right = str.length() - 1;

while (left < right) {

if (str[left] != str[right])

return false;

left++;

right--;

}

return true;

}

int main() {

string input;

cout << "Enter a string: ";

cin >> input;

if (isPalindrome(input))

cout << "The string is a palindrome." << endl;

else

cout << "The string is not a palindrome." << endl;

return 0;

}