REVERSING A NUMBER AND PALINDROME

#include<iostream>

using namespace std;

int main(){

    int reminder;

    int sum=0;

    int temp;

    int n;

    cout << "Enter the number: ";

    cin >> n;

    temp=n;

    while(n>0){

        reminder=n%10;

        sum=(sum\*10)+reminder;

        n=n/10;

    }

    if(sum==temp){

        cout << "Palindrome" << endl;

    }

    else{

        cout << "Not a Palindrome" << endl;

    }

}

REVERSING ARRAY

#include <iostream>

using namespace std;

void reversedArray(int arr2[],int num){

    int start=0;

    int end=num-1;

    int temp;

    while(start<end){

        temp=arr2[start];

        arr2[start]=arr2[end];

        arr2[end]=temp;

        start++;

        end--;

    }

    cout << endl << "Reversed Array" << endl;

    for(int i=0;i<num;i++){

        cout << arr2[i] << " ";

    }

}

int main() {

    int n;

    int arr1[10];

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

    << endl;

    cin >> n;

    cout << "Enter the elements:" << endl;

    for(int i=0;i<n;i++){

        cin >> arr1[i];

    }

    reversedArray(arr1,n);

    return 0;

}

REVERSING STRING

#include<iostream>

using namespace std;

void reverseString(string s1,int l){

    int start=0;

    int end=l-1;

    char temp;

    while(start<end){

        temp=s1[start];

        s1[start]=s1[end];

        s1[end]=temp;

        start++;

        end--;

    }

    cout << endl << "Reversed String:" << endl;

    cout << s1;

}

int main(){

    string s;

    cout << "Enter the string: " << endl;

    cin >> s;

    cout << "Original String:" << endl;

    cout << s << endl;

    int len=s.length();

    cout << "String length: " << len << endl;

    reverseString(s,len);

}

FIBONACCI SERIES

#include<iostream>

using namespace std;

int fib(int n){

    if(n==0 || n==1){

        return 1;

    }

    int num= fib(n-1)+fib(n-2);

    return num;

}

int main(){

    int n;

    cout << "Enter the number: ";

    cin >> n;

    cout << endl << "Fibonacci of a number is"<< endl;

    for(int i=0;i<=n;i++){

        cout << fib(i) << " ";

    }

    return 0;

}

FACTORIAL

#include <iostream>

using namespace std;

int fact(int n);

int main()

{

    int num;

    cout << "Enter the value: " << endl;

    cin >> num;

    int f=fact(num);

    cout << f;

    return 0;

}

int fact(int n){

    if(n==1 || n==0){

        return 1;

    }

    return n\*fact(n-1);

}

CONTAINING DUPLICATES

#include<iostream>

using namespa ce std;

void containsDuplicate(int arr1[],int num) {

    int flag=0;

    for(int i=0;i<num-1;i++){

        for(int j=i+1;j<num;j++){

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

                flag=1;

            }

        }

    }

    if(flag==0){

        cout << "false";

    }

    cout << "true";

}

int main(){

    int n;

    int arr[10];

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

    cin >> n;

    cout << "Enter the elements: " << endl;

    for(int i=0;i<n;i++){

        cin >> arr[i];

    }

    containsDuplicate(arr,n);

}

PRIME NUMBER

#include<iostream>

using namespace std;

int main(){

    int number;

    int flag=0;

    cout << "Enter the number to be checked(Must be greater than or equal to 2): ";

    cin >> number;

    for(int i=2;i<number;i++){

        if(number%i==0){

            flag=1;

        }

    }

    if(flag==0){

        cout << "\nPrime Number";

    }

    else{

        cout << "\nComposite Number";

    }

}

ARMSTRONG NUMBER

#include<iostream>

using namespace std;

int main(){

    int num;

    int rem=0;

    int newnum=0;

    int temp;

    cout << "Enter the number: ";

    cin >> num;

    temp=num;

    while(num>0){

        rem=num%10;

        newnum=newnum+(rem\*rem\*rem);

        num=num/10;

    }

    if(temp==newnum){

        cout << "ARMSTRONG NUMBER";

    }

    else{

        cout << "NOT AN ARMSTRONG NUMBER";

    }

}

SUM OF N NUMBERS

#include<iostream>

using namespace std;

int sumofn(int n){

    if(n==0){

        return 0;

    }

    int sumofnn = n+sumofn(n-1);

    return sumofnn;

}

int main(){

    int n;

    int sum=0;

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

    cin >>n;

    sum=sumofn(n);

    cout << "Sum of natural numbers: " << endl;

    cout <<  sum << endl;

}

FINDING A CHARADTER IN A STRING

1ST WAY WITHOUT USING find FUNCTION

#include<iostream>

using namespace std;

void find(string s,char r){

    int flag=0;

    for(int i=0;i<s.length();i++){

        if(s[i]==r){

            flag=1;

        }

    }

    if(flag==1){

        cout << "Character is found" << endl;

    }

    else{

        cout << "Character is not found" << endl;

    }

}

int main(){

    string s;

    char findingelement;

    cout << "Enter the string: " << endl;

    cin >> s;

    cout << "Enter the element: " << endl;

    cin >> findingelement;

    find(s,findingelement);

}

2ND WAY WITH USE OF find FUNCTION

#include<iostream>

using namespace std;

int main(){

    string s;

    char findingelement;

    int position;

    cout << "Enter the string: " << endl;

    cin >> s;

    cout << "Enter the element: " << endl;

    cin >> findingelement;

    position=s.find(findingelement);

    cout << position << endl;

}

REPEATED CHARACTERS

#include<iostream>

using namespace std;

int count(string s,char c){

    int res=0;

    for(int i=0;i<s.length();i++){

        if(s[i]==c){

            res++;

        }

    }

    return res;

}

int main(){

    string s;

    char element;

    int repeated;

    cout << "Enter the string: " << endl;

    cin >> s;

    cout << "Enter the element: " << endl;

    cin >> element;

    repeated=count(s,element);

    cout << "Number of times the element repeated: " << endl;

    cout << repeated;

}

SUM OF EVEN AND ODD (JAVA LANGUAGE)

import java.util.Scanner;

class Solution {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int evenSum = 0;

        int oddSum = 0;

        while(n > 0) {

            int last = n % 10;

            if(last % 2 == 0) {

                evenSum += last;

            }

            else {

                oddSum += last;

            }

            n = n / 10;

        }

        System.out.println(evenSum + " " + oddSum);

    }

}

SUM OF EVEN NUMBERS TILL N

#include <iostream>

using namespace std;

int main() {

    int n;

    cin >> n;

    int sum=0;

    // for(int i=2;i<=n;i+=2){

    //     sum=sum+i;

    // }

    for(int i=0;i<=n;i++){

        if(i%2==0){

            sum=sum+i;

        }

    }

    cout << sum;

    return 0;

}

FINDING THE POWER OF A NUMBER

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

void power(int p,int n){

    int pow=1;

    for(int i=1;i<=p;i++){

        pow=pow\*n;

    }

    cout << pow;

}

int main() {

    int pow;

    int num;

    cin >> pow;

    cin >> num;

    power(num,pow);

    return 0;

}

COUNTING THE NUMBER OF BITS

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

int countBits(int n){

    int count = 0;

    while (n) {

        count += n & 1;

        n >>= 1;

    }

    return count;

}

int main() {

    int n;

    cin >> n;

    cout<<countBits(n);

    return 0;

}

FINDING THE INDEX OF THE FIRST ELEMENT

#include <iostream>

using namespace std;

// Function for finding first occurrence

void firstIndex(int arr[], int n, int x) {

    int first = -1;

    for (int i = 0; i < n; i++) {

        if (x == arr[i]) {

            first = i;

            break;

        }

    }

    cout << first;

}

int main() {

    int n; //no. of elements

    cin >> n;

    int arr[n];

    //initializing array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    int x; //element whose index to be searched

    cin >> x;

    firstIndex(arr, n, x);

    return 0;

}

FINDING THE INDEX OF THE LAST ELEMENT

#include <iostream>

using namespace std;

// Function for finding last occurrence

void lastIndex(int arr[], int n, int x) {

    int last = -1;

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

        if (x == arr[i]) {

            last = i;

            break;

        }

    }

    cout << last;

}

int main() {

    int n; //no. of elements

    cin >> n;

    int arr[n];

    //initializing array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    int x; //element whose index to be searched

    cin >> x;

    lastIndex(arr, n, x);

    return 0;

}

ROTATION TOWARDS LEFT OF ARRAY

#include <iostream>

using namespace std;

void rotateArray(int arr[], int n, int k) {

    for(int i = 0; i < k; i++) {

        int temp = arr[0];

        for(int j=0;j<n-1;j++){

            arr[j]=arr[j+1];

        }

        arr[n-1]=temp;

    }

    for(int i = 0; i < n; i++){

        cout << arr[i] << " ";

    }

}

int main() {

    int n; // Size of array

    cin >> n;

    int arr[n];

    //initializing array elements

    for(int i=0;i<n;i++){

        cin >> arr[i];

    }

    int k; // No. of times to rotate

    cin >> k;

    rotateArray(arr, n, k);

    return 0;

}

SORT 0 1 AND 2

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

void sort012(int arr[], int n) {

    int countZero = 0;

    int countOne = 0;

    int countTwo = 0;

    for(int i=0;i<n;i++){

        if(arr[i] == 0)

            countZero++;

        else if (arr[i] == 1)

            countOne++;

        else if (arr[i] == 2)

            countTwo++;

    }

    int index = 0;

    for(int i=0;i<countZero;i++){

        arr[index] = 0;

        index++;

    }

    for(int i=0;i<countOne;i++) {

        arr[index] = 1;

        index++;

    }

    for (int i=0;i<countTwo;i++) {

        arr[index] = 2;

        index++;

    }

}

int main() {

    int t;

    cin >> t;

    while (t > 0) {

        int n;

        cin >> n;

        int arr[n];

        for (int i = 0; i < n; i++) {

            cin >> arr[i];

        }

        sort012(arr, n);

        for (int i = 0; i < n; i++) {

            cout << arr[i] << " ";

        }

        cout << endl;

        t--;

    }

    return 0;

}

FINDING NUMBER OF WORDS IN THE STRING

#include <bits/stdc++.h>

#include <iostream>

#include <cstring>

using namespace std;

int countWords(string input) {

    int count=0;

    for(int i=0;i<input[i]!='\0';i++){

        if(input[i]==' '){

            count++;

        }

    }

    return count+1;

}

int main() {

    string input;

    getline(cin, input);

    int output = countWords(input);

    cout << output << endl;

    return 0;

}

BUILT IN FUNCTIONS IN CPP FOR STRING

#include<iostream>

#include<algorithm>

#include <cstring>

using namespace std;

int main(){

//Reverse the string

// string str;

// cout << "Enter the string: ";

// getline(cin,str);

// reverse(str.begin(),str.end());

// cout << "Reversed String using buit in function: "<< str;

// cout << "\n";

// cout << "\n";

//Sort the string

// string str1;

// cout << "Enter the string2: ";

// getline(cin,str1);

// sort(str1.begin(),str1.end());

// cout << "Sorted string using built in function: " << str1;

// cout << "\n";

// cout << "\n";

//Compare the string

// string str2;

// string str3;

// cout << "Enter the string2: ";

// getline(cin,str2);

// cout << "Enter the string3: ";

// getline(cin,str3);

// int res2=strcmp(str2,str3);

// int res1=str2.compare(str3);

// cout << res2;

/\*Retrurns 0 if both the strings are same

Returns <0 if first strings character values is smaller than the second one

Returns >0 if the second string value is greater\*/

// cout << "Compared using compare: " << res1;

//Concatinate the strings

string str1;

string str2;

cout << "Enter the string1: ";

getline(cin,str1);

cout << "Enter the string2: ";

getline(cin,str2);

//string str3=str1+str2;

//cout << "Concatenated string using +: " << str3;

//strcat(str1,str2);

//cout << "Concatenated string using strcat" << str1;

str1.append(str2);

cout << "Concatenated string using append: " << str1;

}

CREATING SUBSTRING

#include <bits/stdc++.h>

#include <iostream>

#include <cstring>

using namespace std;

void printSubstrings(string input) {

    int n = input.size();

    for (int i = 0; i < n; i++) {

        for (int j = i; j < n; j++) {

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

                cout << input[k];

            }

            cout << endl;

        }

    }

}

int main() {

    string input;

    getline(cin, input);

    printSubstrings(input);

    return 0;

}

FINDING CONSECUTIVE DUPLICATES

#include <iostream>

#include <cstring>

using namespace std;

string removeConsecutiveDuplicates(string input)

{

    int n = input.size();

    if (n == 0)

    {

        return input;

    }

    string answer;

    answer += (input[0]);

    int i = 1;

    while (i < n)

    {

        if (input[i] != input[i - 1])

        {

            answer += (input[i]);

        }

        i += 1;

    }

    return answer;

}

int main()

{

    string str;

    cin >> str;

    string ans = removeConsecutiveDuplicates(str);

    cout << ans << endl;

}

REVERSE EACH WORDS IN A STRING

#include <iostream>

#include <vector>

using namespace std;

string reverseStringWordWise(string input)

{

    vector<string> words;

    int n = input.size();

    string word;

    // Extract each word

    for (int i = 0; i < n; i++)

    {

        if (input[i] == ' ')

        {

            words.push\_back(word);

            word = "";

        }

        else

        {

            word.push\_back(input[i]);

        }

    }

    words.push\_back(word);

    // Reverse the words in the sentence

    int i = 0, j = words.size() - 1;

    while (i < j)

    {

        swap(words[i], words[j]);

        i++;

        j--;

    }

    // Append back the words

    string ans;

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

    {

        ans.append(words[i]);

        ans.push\_back(' ');

    }

    return ans;

}

int main()

{

    string s;

    getline(cin, s);

    string ans = reverseStringWordWise(s);

    cout << ans << endl;

}

CHECK PERMUTATION OF THE STRING

#include <iostream>

#include <cstring>

using namespace std;

bool isPermutation(char input1[], char input2[]) {

    int frequency[256];

    for (int i = 0; i < 256; ++i) {

        frequency[i] = 0;

    }

    for (int i = 0; input1[i] != '\0'; ++i) {

        ++frequency[input1[i]];

    }

    for (int i = 0; input2[i] != '\0'; ++i) {

        --frequency[input2[i]];

    }

    for (int i = 0; i < 256; ++i) {

        if (frequency[i] != 0) {

            return false;

        }

    }

    return true;

}

int main() {

    int size = 1e6;

    char str1[size];

    char str2[size];

    cin >> str1 >> str2;

    cout << (isPermutation(str1, str2) ? "true" : "false");

}