1. **Write a program to read in two integers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.**

#include <iostream>

using namespace std;

int main(){

int num1,num2;

cout<<"enter the first number";

cin>>num1;

cout<<"enter the second number";

cin>>num2;

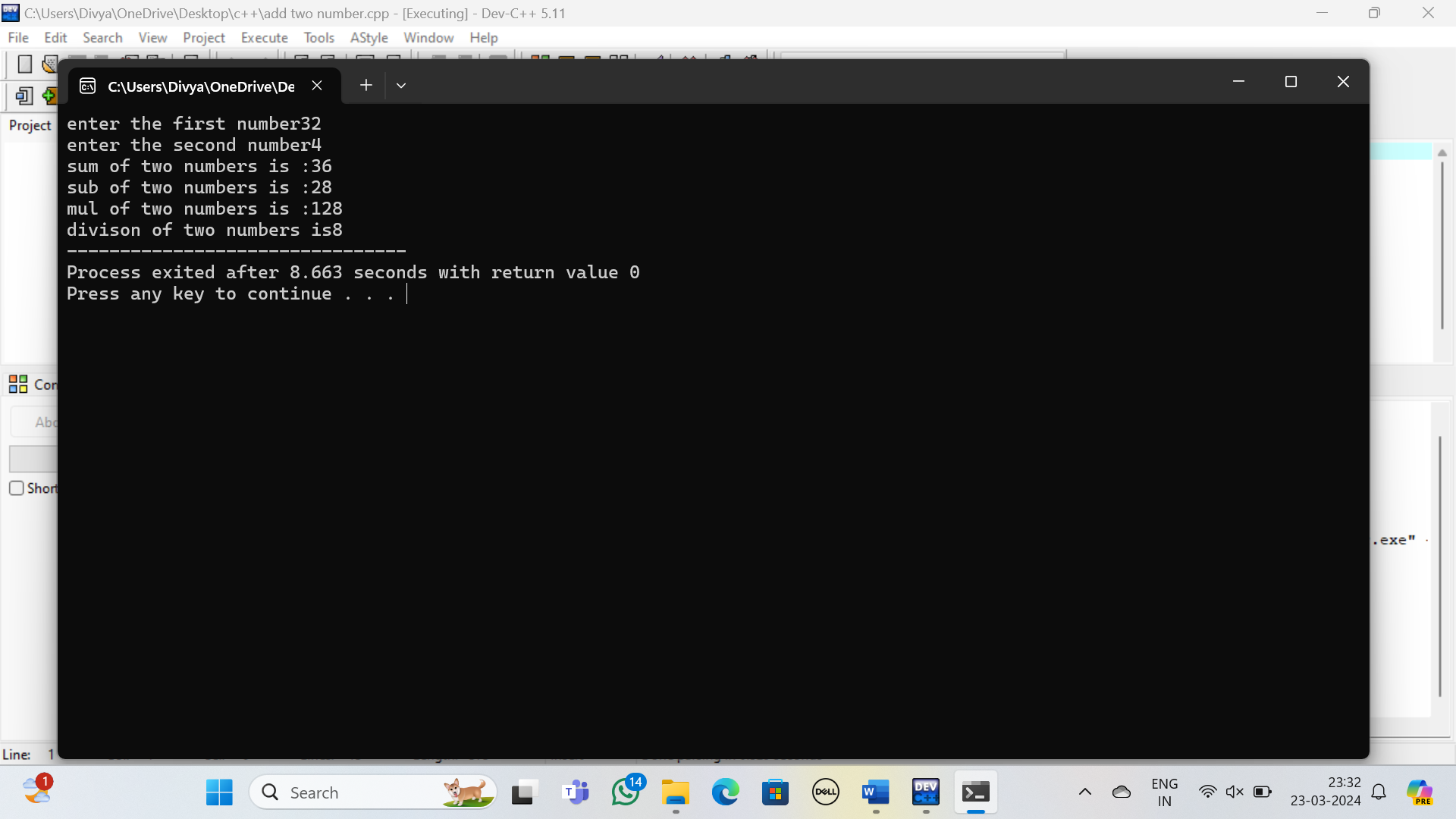
cout<<"sum of two numbers is :"<<num1+num2<<endl;

cout<<"sub of two numbers is :"<<num1-num2<<endl;

cout<<"mul of two numbers is :"<<num1\*num2<<endl;

cout<<"divison of two numbers is"<<num1/num2;

}



1. **Program to determine the integer is odd or even**

#include <iostream>

using namespace std;

int main(){

int num;

cout<<"enter the given number:";

cin>>num;

if(num%2==0){

cout<<num<<" is a even";

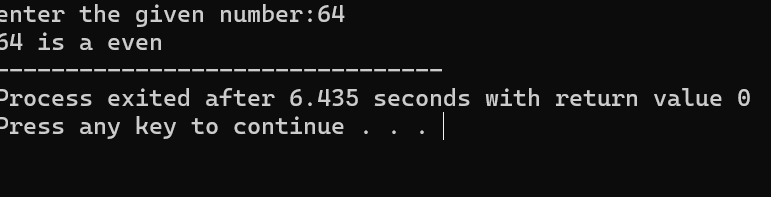
}

else{

cout<<num<<"is a odd";

}

}

****

1. **Program to compute the average of three integers**

#include<iostream>

using namespace std;

int main(){

int num1,num2,num3;

double avg;

cout<<"enter the first interger:";

cin>>num1;

cout<<"enter the second interger:";

cin>>num2;

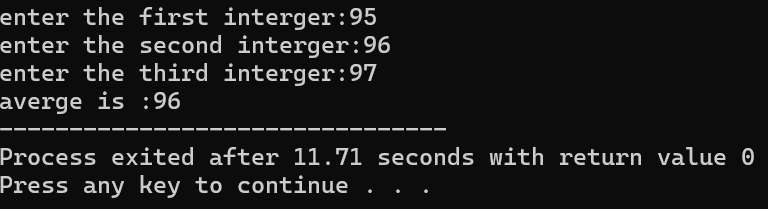
cout<<"enter the third interger:";

cin>>num3;

avg=(num1+num2+num3)/3;

cout<<"averge is :"<<avg;

}



1. **Program to check two numbers are equal or not**

#include<iostream>

using namespace std;

int main(){

string str,str1;

cout<<"enter the first string:";

cin>>str;

cout<<"enter the second string:";

cin>>str1;

cout<<n;

if(str==str1){

cout<<" two strings are equal:";

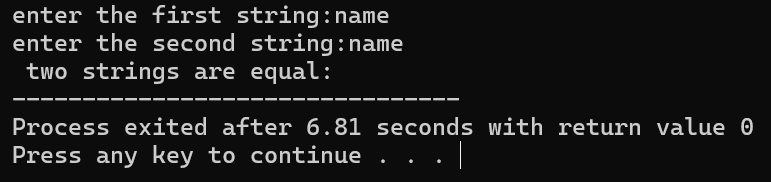
}

else{

cout<<"two strings are not equal";

}

}



1. **Write a program to read in two Floating numbers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.**

#include <iostream>

using namespace std;

int main(){

float num1,num2;

cout<<"enter the first number";

cin>>num1;

cout<<"enter the second number";

cin>>num2;

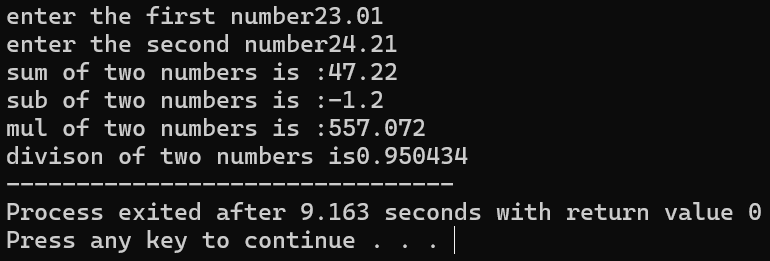
cout<<"sum of two numbers is :"<<num1+num2<<endl;

cout<<"sub of two numbers is :"<<num1-num2<<endl;

cout<<"mul of two numbers is :"<<num1\*num2<<endl;

cout<<"divison of two numbers is"<<num1/num2;

}

****

1. **Program to check the character is a vowel or consonant**

#include<iostream>

using namespace std;

int main(){

char ch;

cout<<"enter the given charcter:";

cin>>ch;

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

cout<<ch<<"is a vowel";

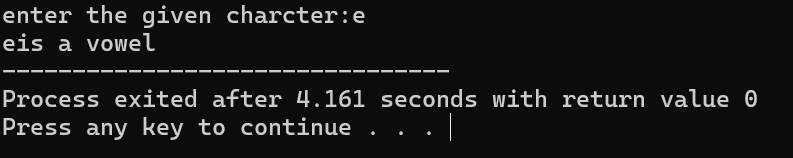
}

else{

cout<<" is consonet";

}

}



1. **Program to check the number is positive, negative or zero**

#include <iostream>

using namespace std;

int main(){

int num;

cout<<"enter the given number:";

cin>>num;

if(num<0){

cout<<num<<" is a negtive number:";

}

else if(num>0){

cout<<num<<"is a postive number:";

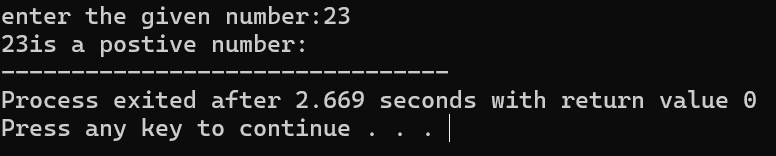
}

else{

cout<<num<<"is a equal to zero";

}

}



1. **Program to determine which number is greater among two integer**

#include<iostream>

using namespace std;

int main(){

int num ,num1;

cout<<"enter the first number:";

cin>>num;

cout<<"enter the second number:";

cin>>num1;

if(num>num1){

cout<<num<<"is greater than "<< num1;

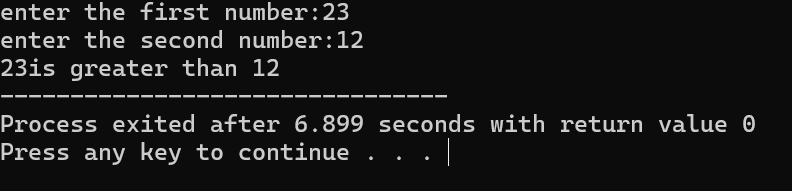
}

else{

cout<<num1<<"is greater than "<< num;

}

}



1. **Program to read a floating-number and round it to the nearest integer using the floor an ceil functions.**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

float num;

cout << "Enter the given float number: ";

cin >> num;

int n = floor(num);

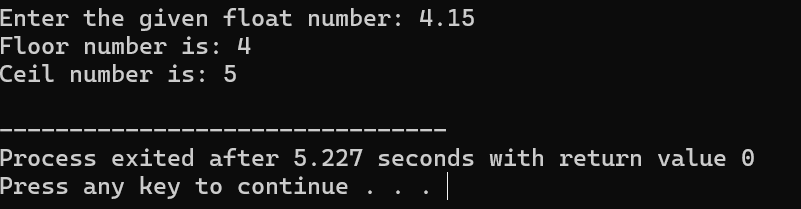
int m = ceil(num);

cout << "Floor number is: " << n << endl;

cout << "Ceil number is: " << m << endl;

return 0;

}

****

1. **Program to swap two numbers using bitwise XOR operator**

#include <iostream>

using namespace std;

void swapNumbers(int &a, int &b) {

a = a ^ b;

b = a ^ b;

a = a ^ b;

}

int main() {

int num1, num2;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

cout << "Before swapping: " << endl;

cout << "First number: " << num1 << endl;

cout << "Second number: " << num2 << endl;

swapNumbers(num1, num2);

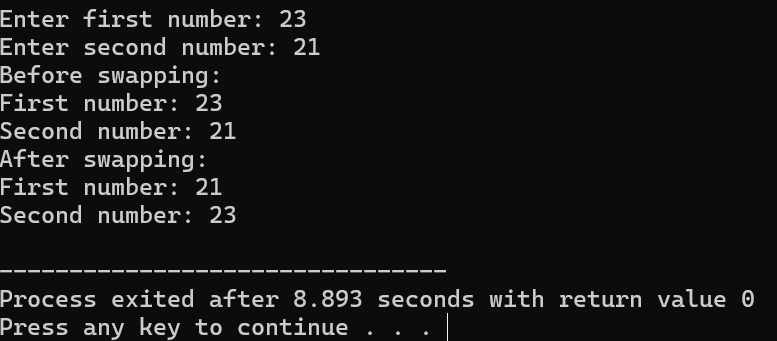
cout << "After swapping: " << endl;

cout << "First number: " << num1 << endl;

cout << "Second number: " << num2 << endl;

return 0;

}



1. **Largest among three numbers using ternary conditional operator**

#include<iostream>

using namespace std;

int main() {

int num1, num2, num3;

cout << "Enter the first number:";

cin >> num1;

cout << "Enter the second number:";

cin >> num2;

cout << "Enter the third number:";

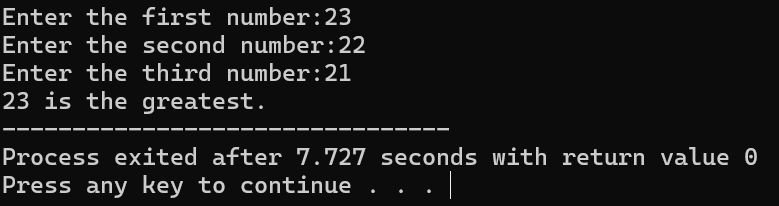
cin >> num3;

int greatest = (num1 > num2 ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3));

cout << greatest << " is the greatest.";

return 0**;**

**}**

****

1. **Program to check the integer is divisible by 3 or not using ternary conditional operator**

#include<iostream>

using namespace std;

int main() {

int num;

cout << "Enter a number: ";

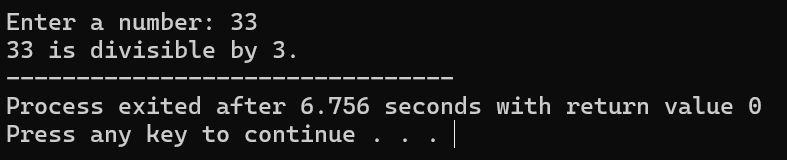
cin >> num;

string divisibleBy3 = (num % 3 == 0) ? " is divisible by 3." : " is not divisible by 3.";

cout << num << divisibleBy3;

return 0;

}

****

1. **Program to print numbers from 1 to 10 using for loop**

#include<iostream>

using namespace std;

int main(){

int num,sum=0;

cout<<"enter the limit:";

cin>>num;

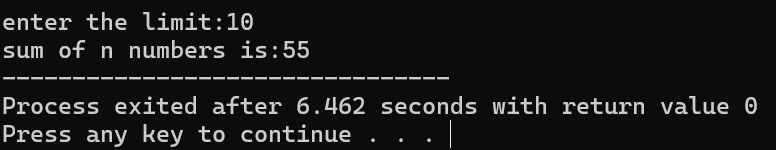
for(int i=1;i<=10;i++){

sum=sum+i;

}

cout<<"sum of n numbers is:"<<sum;

}

****

1. **Factorial of a number using for loop**

#include<iostream>

using namespace std;

int main(){

int num,fact=1;

cout<<"enter the given number:";

cin>>num;

if(num<0){

cout<<"negtive number we can't find facorial";

}

else if(num==0 || num==1){

cout<<"factorial of zero or one is:"<<1;

}

else{

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

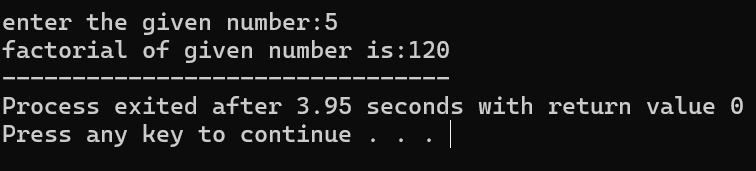
fact=fact\*i;

}

cout<<"factorial of given number is:"<<fact;

}

}



1. **Print multiplication table using for loop**

#include<iostream>

using namespace std;

int main(){

int table,display;

cout<<"enter which table you want to display:";

cin>>table;

cout<<"enter upto which element we want to display:";

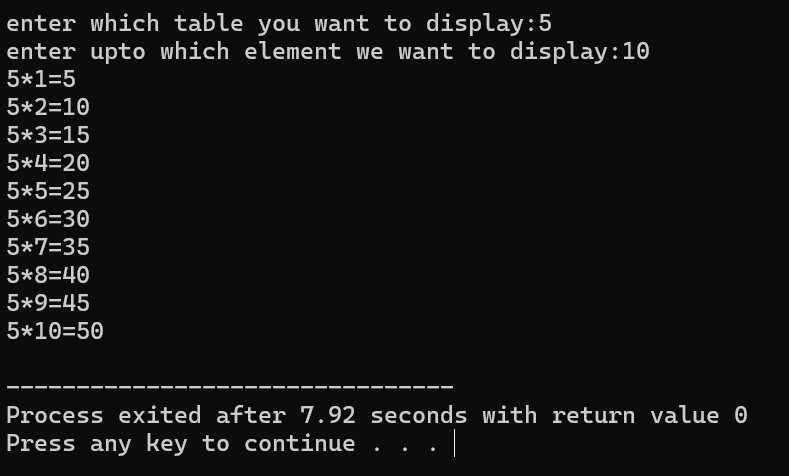
cin>>display;

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

cout<<table<<"\*"<<i<<"="<<table\*i<<endl;

}

}



1. **Fibonacci series using for loop**

#include <iostream>

using namespace std;

int main() {

int num, num1, num2, num3;

cout << "Enter number of terms to display: ";

cin >> num;

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

if (num <= 0) {

cout << "Invalid input";

} else if (num == 1) {

cout << num1;

} else if (num == 2) {

cout << num1 << " " << num2;

} else {

cout << num1 << " " << num2 << " ";

for (int i = 3; i <= num; ++i) {

num3 = num1 + num2;

cout << num3 << " ";

num1 = num2;

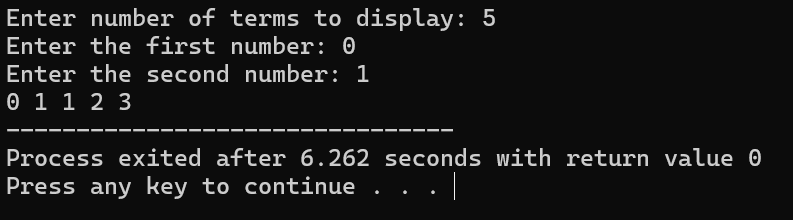
num2 = num3;

}

}

return 0;

}



1. **Prime number using for loop**

#include<iostream>

using namespace std;

int main(){

int num;

cout<<"enter the given number:";

cin>>num;

bool isprime=true;

for(int i=2;i<num;i++){

if(num%i==0){

isprime=false;

break;

}

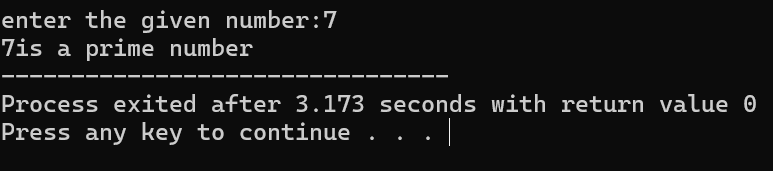
}

if(isprime){

cout<<num<<"is a prime number";

}

}



1. **Check the string is palindrome or not using while loop**

#include<iostream>

using namespace std;

int main(){

string str,str1="";

cout<<"enter the given string:";

cin>>str;

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

str1=str1+str[i];

}

cout<<"reverse string is: "<<str1<<endl;

if(str==str1){

cout<<"string is palindrome";

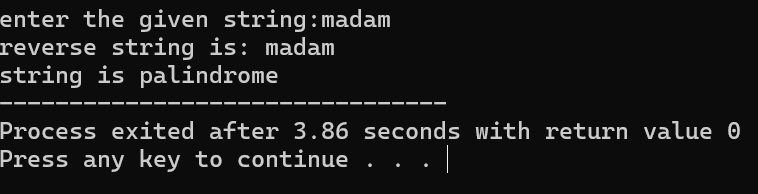
}

else{

cout<<"string is not palindrome";

}

}



1. **Sum of all digits using while loop (n=123 output:1+2+3=6) 21.**

#include<iostream>

using namespace std;

int main(){

int num,sum=0;

cout<<"enter the given number";

cin>>num;

while(num!=0){

int rem=num%10;

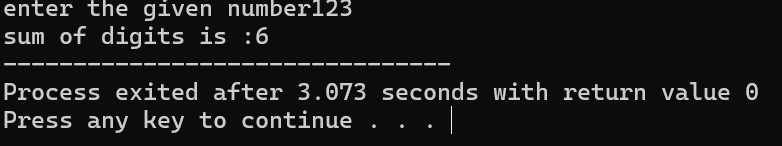
sum=sum+rem;

num=num/10;

}

cout<<"sum of digits is :"<<sum;

}



1. **GCD of two numbers using do-while loop**

#include <iostream>

using namespace std;

int main() {

int num1, num2;

cout << "Enter first number: ";

cin >> num1;

cout << "Enter second number: ";

cin >> num2;

int remainder;

int gcd;

do {

remainder = num1 % num2;

if (remainder == 0) {

gcd = num2;

} else {

num1 = num2;

num2 = remainder;

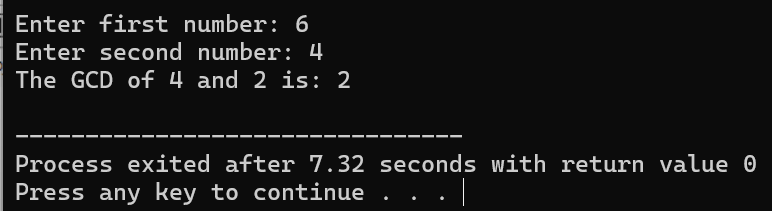
}

} while (remainder != 0);

cout << "The GCD of " << num1 << " and " << num2 << " is: " << gcd << endl;

return 0;

}



1. **Check whether the number is perfect or not**

#include<iostream>

using namespace std;

int main(){

int num,sum=0;

cout<<"enter the given number:";

cin>>num;

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

if(num%i==0){

sum=sum+i;

}

}

if(sum==num){

cout<<"is a perfect number";

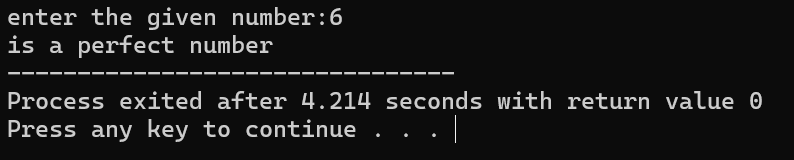
}

else{

cout<<"is not a perfect number";

}

}



1. **Armstrong number**

#include<iostream>

#include<cmath>

using namespace std;

int main(){

int num,temp,rem,sum=0;

cout<<"enter the given number:";

cin>>num;

temp=num;

while(num!=0){

rem=num%10;

sum=pow(rem ,3)+sum;

num=num/10;

}

if(sum==temp){

cout<<"is amstrong number";

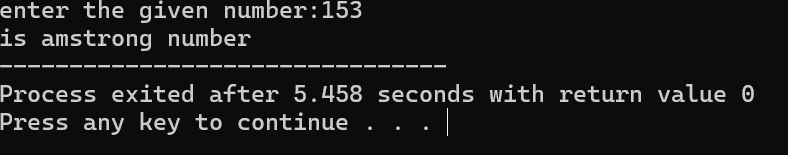
}

else{

cout<<"is not a amstrong number";

}

}



1. **Harshad number**

#include<iostream>

using namespace std;

int main(){

int num,sum=0;

cout<<"enter the given number:";

cin>>num;

while(num!=0){

int rem=num%10;

sum=sum+rem;

num=num/10;

}

if(num%sum==0){

cout<<"is a harshad number";

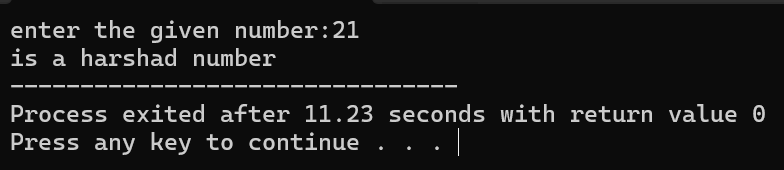
}

else{

cout<<"is not a harshad number";

}

}



1. **Happy number**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

int num, sum = 0;

cout << "Enter the given number: ";

cin >> num;

while (num != 1 && num != 4) {

sum = 0;

while (num > 0) {

int rem = num % 10;

sum += pow(rem, 2);

num /= 10;

}

num = sum;

}

if (num == 1) {

cout << "It is a happy number";

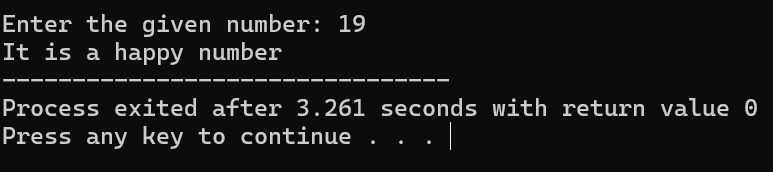
} else {

cout << "It is not a happy number";

}

return 0;

}



1. **strong number**

#include<iostream>

using namespace std;

int main(){

int num,fact=1,temp,sum=0;

cout<<"enter the given number:";

cin>>num;

temp=num;

while(num!=0){

int rem=num%10;

fact=1;

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

fact=fact\*i;

}

sum=sum+fact;

num=num/10;

}

if(temp==sum){

cout<<"is a strong number:";

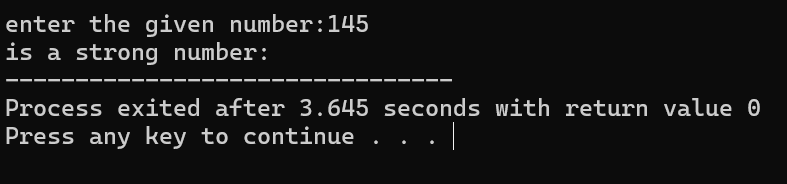
}

else{

cout<<"is not a strong number";

}

}



1. **buzz number**

#include<iostream>

using namespace std;

int main(){

int num;

cout<<"enter the given number:";

cin>>num;

if(num%7==0 || num%10==7){

cout<<"is a buzz number";

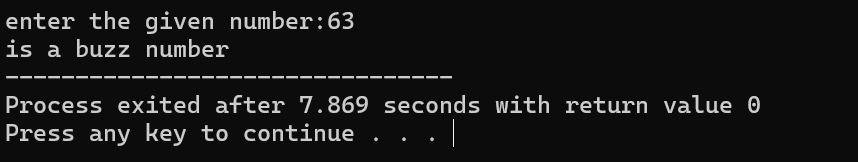
}

else{

cout<<"is not a buzz number";

}

}



1. **neon number**

#include<iostream>

#include<cmath>

using namespace std;

int main(){

int num ,temp,sum=0;

cout<<"enter the given number:";

cin>>num;

temp=pow(num,2);

while(temp!=0){

int rem=temp%10;

sum=sum+rem;

temp=temp/10;

}

if(sum==num){

cout<<num<<"is a neon number:";

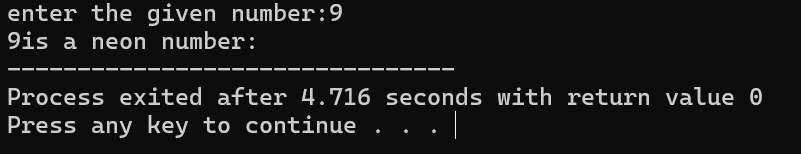
}

else{

cout<<num<<"is not a neon number:";

}

}



**29. abundant number**

#include<iostream>

using namespace std;

int main(){

int num,sum=0;

cout<<"enter the given number:";

cin>>num;

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

if(num%i==0){

sum=sum+i;

}

}

if(sum>num){

cout<<"is a abudent number";

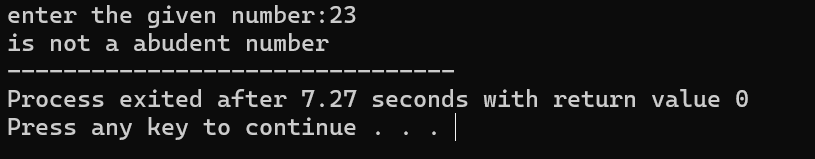
}

else{

cout<<"is not a abudent number";

}

}



**30. narcissistic number**

#include<iostream>

#include<cmath>

using namespace std;

int main(){

int num,temp,rem,sum=0;

cout<<"enter the given number:";

cin>>num;

temp=num;

while(num!=0){

rem=num%10;

sum=pow(rem ,3)+sum;

num=num/10;

}

if(sum==temp){

cout<<"is amstrong number";

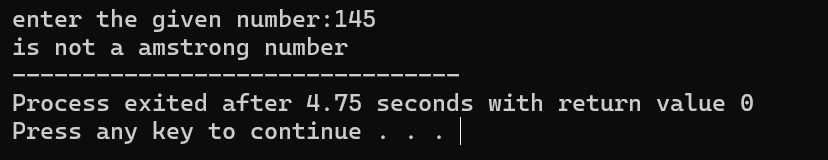
}

else{

cout<<"is not a amstrong number";

}

}



**31. print the pattern 1 22 333 4444 55555**

#include<iostream>

using namespace std;

int main(){

int rows;

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

cin>>rows;

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

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

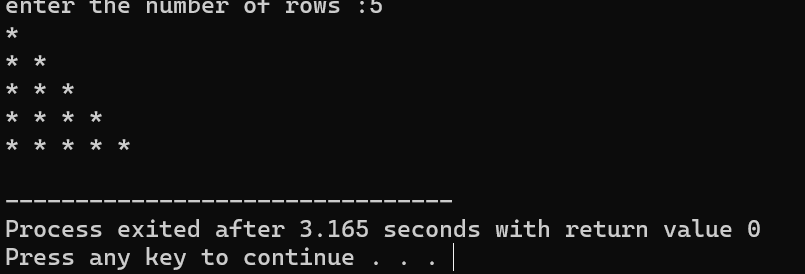
cout<<"\*"<<" ";

}

cout<<endl;

}

}



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

#include<iostream>

using namespace std;

int main(){

int rows;

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

cin>>rows;

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

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

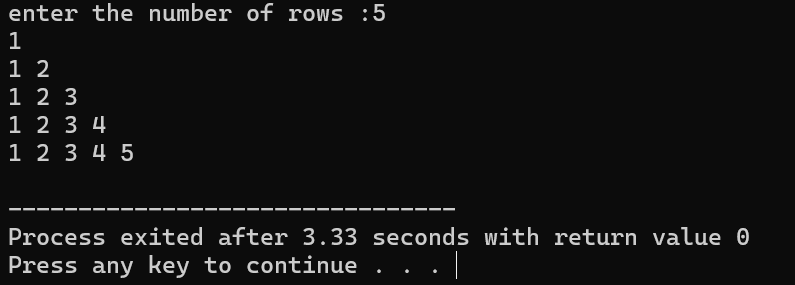
cout<<j<<" ";

}

cout<<endl;

}

}



**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++){

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

cout<<" ";

}

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

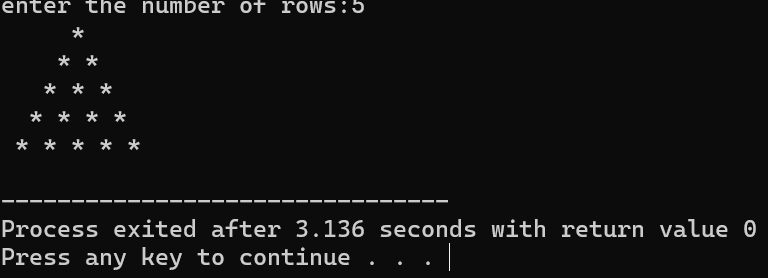
cout<<"\*"<<" ";

}

cout<<endl;

}

}



**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;

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

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

cout<<" ";

}

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

cout<<"\*"<<" ";

}

cout<<endl;

}

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

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

cout<<" ";

}

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

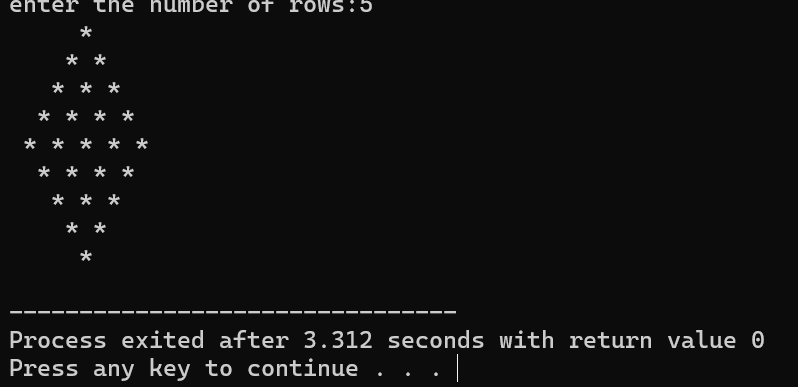
cout<<"\*"<<" ";

}

cout<<endl;

}

}



**35. Program to reverse the elements in the array**

#include<iostream>

using namespace std;

int main(){

int n;

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

cin>>n;

int a[n];

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

cout<<"enter the "<<"element"<<" "<<i+1<<":";

cin>>a[i];

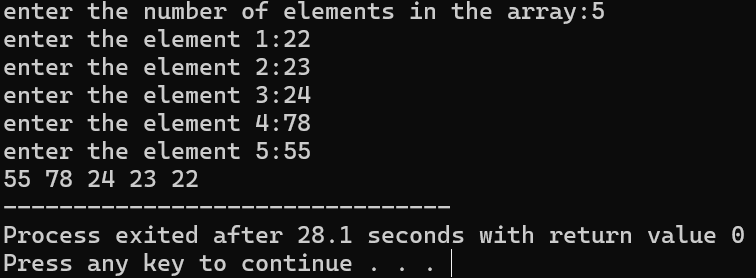
}

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

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

}

}



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

#include<iostream>

using namespace std;

int main(){

int n;

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

cin>>n;

int a[n];

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

cout<<"enter the "<<"element"<<" "<<i+1<<":";

cin>>a[i];

}

cout<<"before inserting element array is :";

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

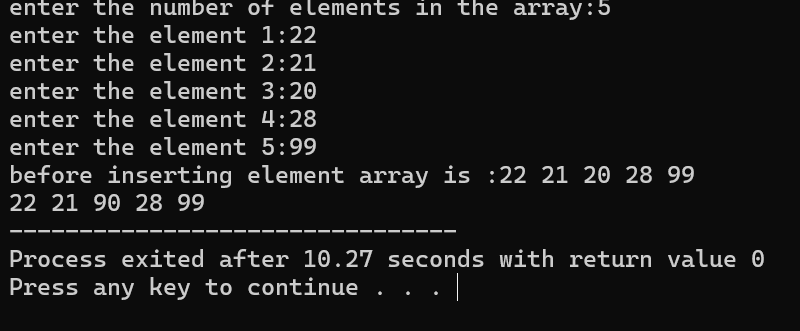
cout<<a[i]<<" ";

}

cout<<endl;

a[2]=90;

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

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

}

}

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

#include <iostream>

using namespace std;

int main() {

const int maxSize = 100; // Maximum size of the array

int arr[maxSize]; // Declare the array

int n; // Number of elements in the array

// Input the number of elements

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

cin >> n;

// Input the elements

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

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

cin >> arr[i];

}

int positionToDelete;

cout << "Enter the position of the element to delete (0-indexed): ";

cin >> positionToDelete;

// Check if the position is valid

if (positionToDelete >= 0 && positionToDelete < n) {

// Shift elements to the left to overwrite the deleted element

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

arr[i] = arr[i + 1];

}

// Decrement the number of elements

--n;

// Print the updated array

cout << "Updated array after deletion: ";

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

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

}

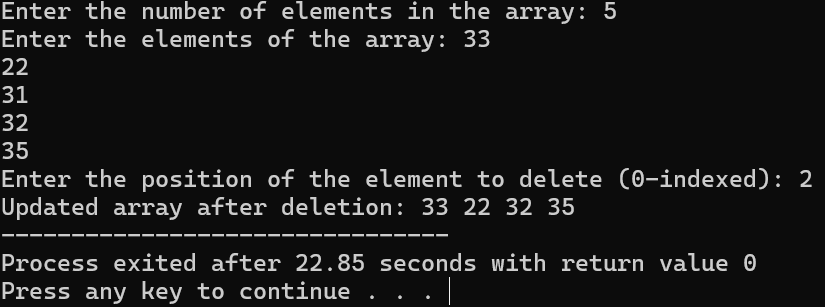
} else {

cout << "Invalid position!";

}

return 0;

}

****

**38. Find the sum of all elements in an array**

#include<iostream>

using namespace std;

int main(){

int n,sum=0;

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

cin>>n;

int a[n];

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

cout<<"enter the "<<"element"<<" "<<i+1<<":";

cin>>a[i];

}

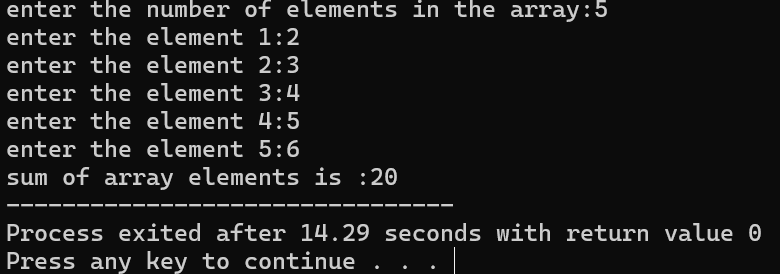
for(int i=0;i<n ;i++){

sum=sum+a[i];

}

cout<<"sum of array elements is :"<<sum;

}



**39. Find the average of all elements in an array**

#include<iostream>

using namespace std;

int main(){

int n,sum=0;

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

cin>>n;

int a[n];

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

cout<<"enter the "<<"element"<<" "<<i+1<<":";

cin>>a[i];

}

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

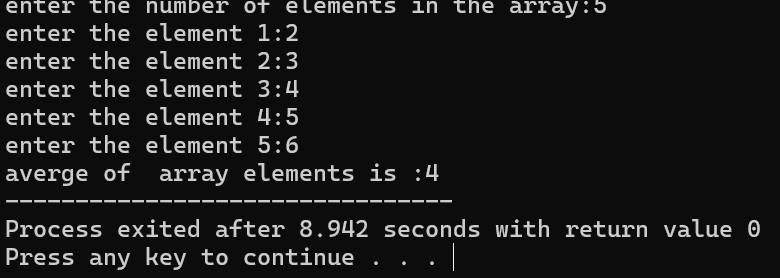
sum=sum+a[i];

}

float avg=sum/n;

cout<<"averge of array elements is :"<<avg;

}



**40. Find the second largest element in an array**

**#**include<iostream>

#include<algorithm>

using namespace std;

int main(){

int n,m,k;

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

cin>>n;

int a[n];

cout<<"mth smallest number:";

cin>>m;

cout<<"kth largest number:";

cin>>k;

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

cout<<"enter the "<<"element"<<" "<<i+1<<":";

cin>>a[i];

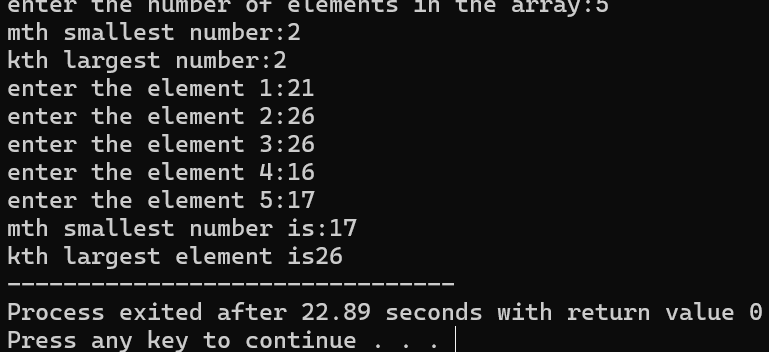
}

sort(a,a+n);

cout<<"mth smallest number is:"<<a[m-1]<<endl;

cout<<"kth largest element is"<<a[n-k];

}



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

#include<iostream>

using namespace std;

int main(){

int n ,k,count=0;

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

cin>>n;

cout<<"which element:";

cin>>k;

int a[n];

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

cout<<"enter the element"<<(i+1)<<":";

cin>>a[i];

}

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

if(a[i]==k){

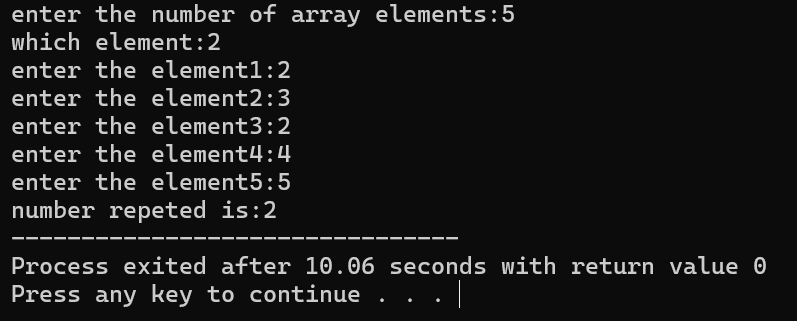
count+=1;

}

}

cout<<"number repeted is:"<<count;

}



42. Merge two array

#include<iostream>

using namespace std;

int main() {

int n, k;

cout << "Enter the number of elements in array1: ";

cin >> n;

cout << "Enter the number of elements in array2: ";

cin >> k;

int a[n];

int b[k];

int c[n + k];

cout << "Enter elements for array1:\n";

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

cout << "Enter the element " << (i + 1) << ": ";

cin >> a[i];

}

cout << "Enter elements for array2:\n";

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

cout << "Enter the element " << (i + 1) << ": ";

cin >> b[i];

}

cout << endl;

cout << "Array1 elements are: ";

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

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

}

cout << endl;

cout << "Array2 elements are: ";

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

cout << b[i] << " ";

}

cout << endl;

cout << "Merge of two arrays is: ";

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

c[i] = a[i];

cout << c[i] << " ";

}

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

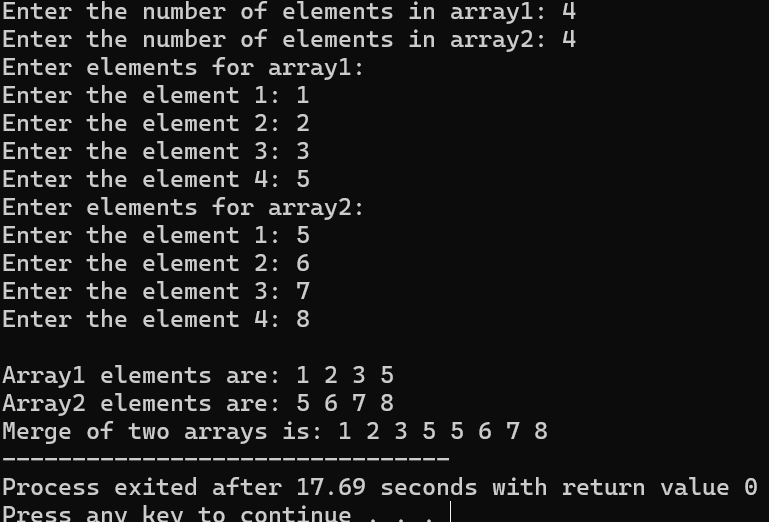
c[n + i] = b[i];

cout << c[n + i] << " ";

}

return 0;

}



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

#include <iostream>

using namespace std;

int main() {

int size;

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

cin >> size;

// Dynamically allocate memory for the array

int \*dynamicArray = new int[size];

// Input values into the dynamic array

cout << "Enter " << size << " values for the dynamic array:\n";

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

cout << "Enter value " << (i + 1) << ": ";

cin >> dynamicArray[i];

}

// Display the values of the dynamic array

cout << "Values of the dynamic array are: ";

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

cout << dynamicArray[i] << " ";

}

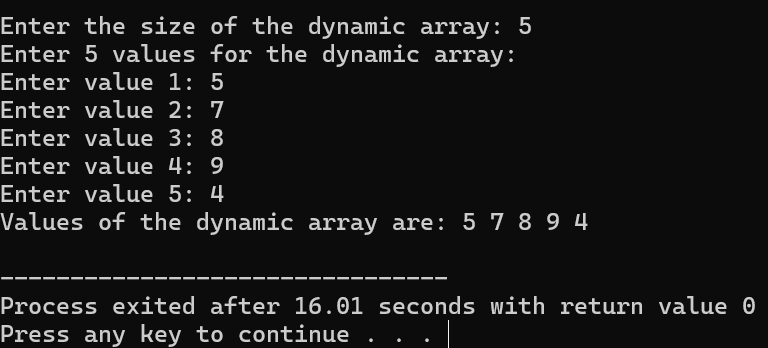
cout << endl;

// Free the dynamically allocated memory

delete[] dynamicArray;

return 0;

}



45. Add 2 matrices

#include<iostream>

using namespace std;

int main(){

int rows,cols,i,j;

cout<<"enter the no.of.rows:";

cin>>rows;

cout<<"enter the no.of.cols:";

cin>>cols;

int a[rows][cols];

int b[rows][cols];

int c[rows][cols];

cout<<"enter the first matrix elements:"<<endl;

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

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

cout<<"enter the element"<<(i+1)<<(j+1)<<":";

cin>>a[i][j];

}

}

cout<<"second matrix elements:"<<endl;

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

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

cout<<"enter the element"<<(i+1)<<(j+1)<<":";

cin>>b[i][j];

}

}

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

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

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

}

}

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

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

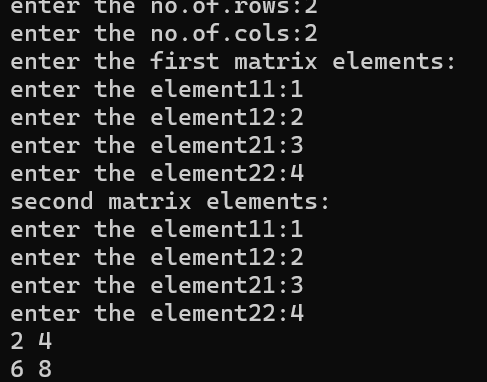
cout<<c[i][j]<<" ";

}

cout<<endl;

}

}



**46. Multiply 2 matrices**

#include<iostream>

using namespace std;

int main(){

int a[2][2]={{1,2},{1,2}};

int b[2][2]={{1,2},{2,1}};

int c[2][2];

cout<<"mul matrix :"<<endl;

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

for(int j=0;j<2;j++){

c[i][j]=0;

for(int k=0 ;k<2;k++){

c[i][j]+=a[i][k]\*b[k][j];

}

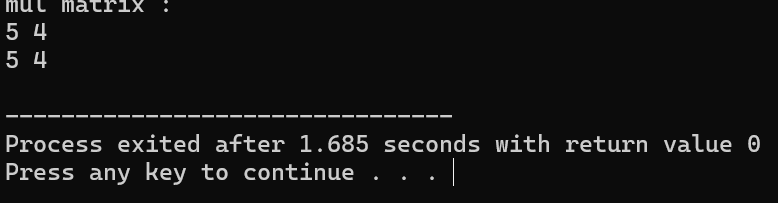
cout<<c[i][j]<<" ";

}

cout<<endl;

}

}



47. Find the sum of diagonals of a matrix

#include<iostream>

using namespace std;

int main() {

int matrix[3][3] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

int rows = 3;

int cols = 3;

int sum\_main\_diag = 0;

int sum\_secondary\_diag = 0;

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

sum\_main\_diag += matrix[i][i];

}

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

sum\_secondary\_diag += matrix[i][cols - 1 - i];

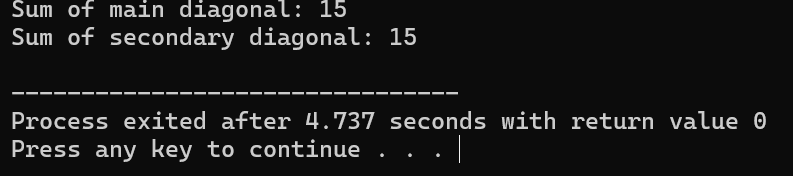
}

cout << "Sum of main diagonal: " << sum\_main\_diag << endl;

cout << "Sum of secondary diagonal: " << sum\_secondary\_diag << endl;

return 0;

}



**Functions in C++**

**48.Find factorial using function**

#include<iostream>

using namespace std;

int factorial(int n){

int fact=1;

if(n<=0){

return -1;

}

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

return 1;

}

else{

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

fact=fact\*i;

}

return fact;

}

}

int main()

{

int n;

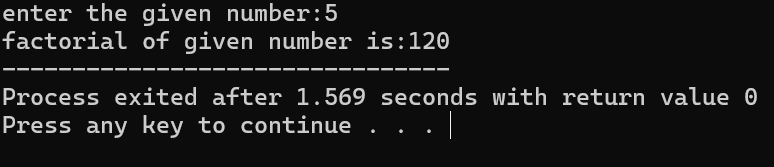
cout<<"enter the given number:";

cin>>n;

int result=factorial(n);

cout<<"fact is"<<result;

}



**49. Find 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;

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

is\_prime = false;

}

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

if (n % i == 0) {

is\_prime = false;

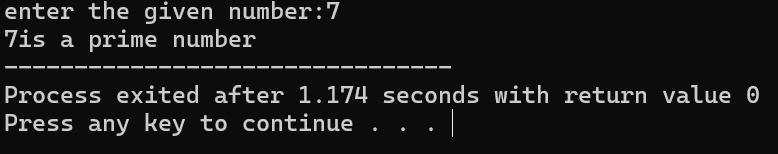
break;

}

}

return is\_prime;

**}**

****

**50. Find the reverse of a string using function**

#include<iostream>

using namespace std;

void reverse(string str){

int n= str.length();

string str1="";

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

str1=str[i]+str1;

}

cout<<"reverse string is:"<<str1;

}

int main(){

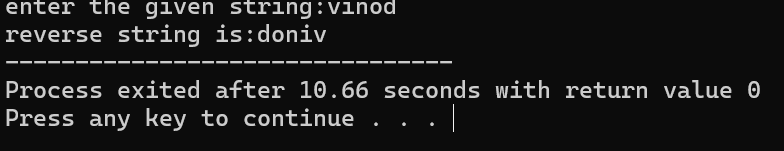
string str;

cout<<"enter the given string:";

cin>>str;

reverse(str);

}



51. Find minimum and maximum element in an array using function

#include<iostream>

#include<algorithm>

using namespace std;

void findMinMax(int a[], int n) {

sort(a, a + n);

cout << "Min element is: " << a[0] << endl;

cout << "Max element is: " << a[n - 1] << endl;

}

int main() {

int n;

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

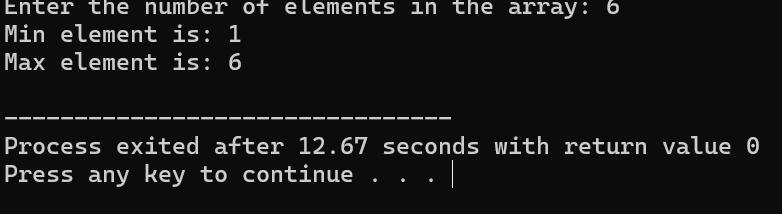
cin >> n;

int a[n] = {3, 2, 4, 5, 1, 6};

findMinMax(a, n);

return 0;

}



**52. Find GCD of two number using function**

#include <iostream>

using namespace std;

int gcd(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int main() {

int num1, num2;

cout << "Enter the first number: ";

cin >> num1;

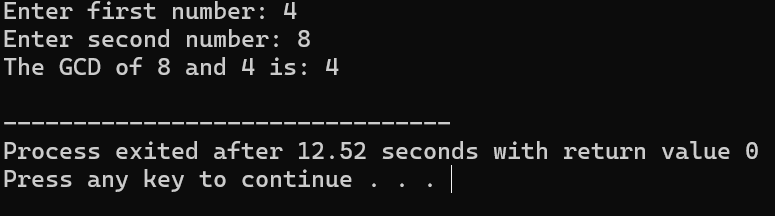
cout << "Enter the second number: ";

cin >> num2;

cout << "GCD of " << num1 << " and " << num2 << " is: " << gcd(num1, num2) << endl;

return 0;

}



**53. Function to count the no of elements in a string**

#include <iostream>

using namespace std;

// Function to count the number of elements in a string

int countElements(const string& str) {

int count = 0;

for (char ch : str) {

count++;

}

return count;

}

int main() {

string str;

cout << "Enter a string: ";

getline(cin, str); // Using getline to read the entire line including spaces

int elementCount = countElements(str);

cout << "Number of elements in the string: " << elementCount << endl;

return 0;

}

**54. Convert Celsius and Fahrenheit using function**

#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 << celsius << " Celsius is equal to " << fahrenheit << " Fahrenheit." << endl;

cout << "Enter temperature in Fahrenheit: ";

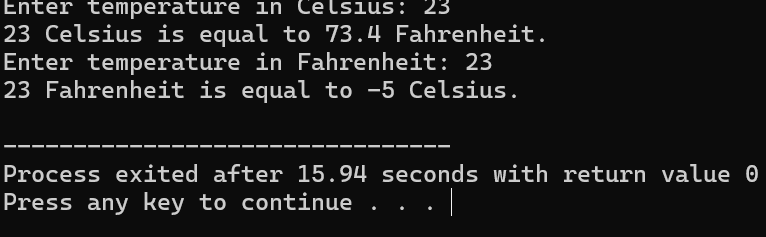
cin >> fahrenheit;

celsius = fahrenheitToCelsius(fahrenheit);

cout << fahrenheit << " Fahrenheit is equal to " << celsius << " Celsius." << endl;

return 0;

}

****

55**. Find the area of a circle using function**

#include <iostream>

using namespace std;

const double PI = 3.14159; // Define PI as a constant

// Function to calculate the area of a circle

double calculateArea(double radius) {

return PI \* radius \* radius;

}

int main() {

double radius;

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

cin >> radius;

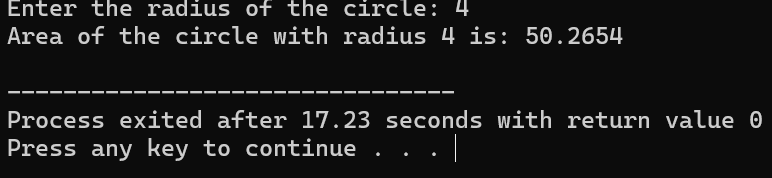
// Calculate the area using the function and display the result

double area = calculateArea(radius);

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

return 0;

}



**56. Check whether the string is palindrome or not**

#include <iostream>

#include <string>

#include <algorithm>

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 str;

cout << "Enter a string: ";

cin >> str;

transform(str.begin(), str.end(), str.begin(), ::tolower);

if (isPalindrome(str)) {

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

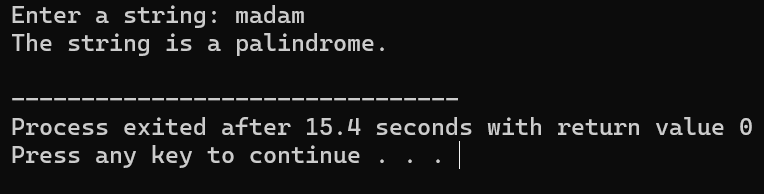
} else {

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

}

return 0;

}



**57. Write a c++ program to create a class for a bank account with a constructor and a destructor**

#include <iostream>

class BankAccount {

private:

int accountNumber;

double balance;

public:

BankAccount(int accNum, double initialBalance) : accountNumber(accNum), balance(initialBalance) {

std::cout << "Account created with Account Number: " << accountNumber << std::endl;

}

~BankAccount() {

std::cout << "Account with Account Number: " << accountNumber << " has been closed." << std::endl;

}

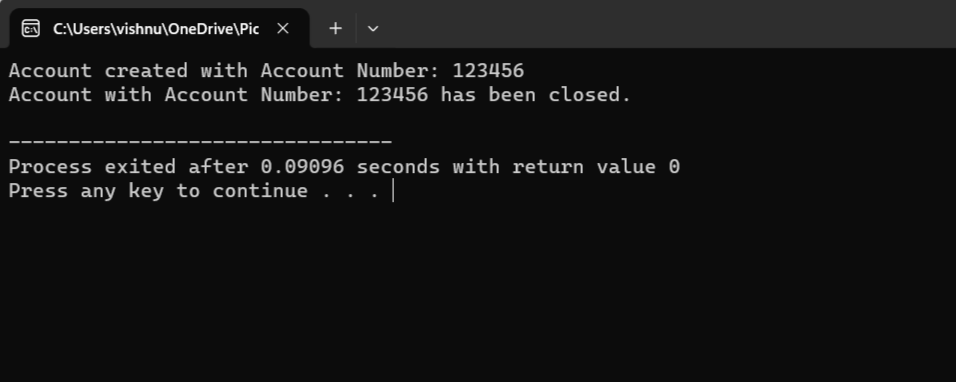
};

int main() {

BankAccount myAccount(123456, 1000.0);

return 0;

}



**58.** **Write a c++ program to create a class for a car with a constructor and a destructor**

#include <iostream>

class Car {

private:

std::string brand;

std::string model;

public:

Car(std::string carBrand, std::string carModel) : brand(carBrand), model(carModel) {

std::cout << "A new car of brand " << brand << " and model " << model << " has been created." << std::endl;

}

~Car() {

std::cout << "The car of brand " << brand << " and model " << model << " has been destroyed." << std::endl;

}

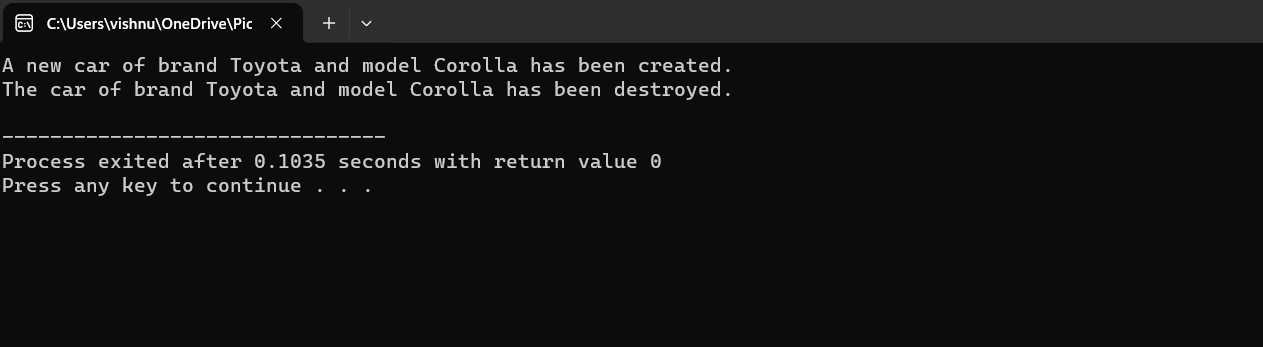
};

int main() {

Car myCar("Toyota", "Corolla");

return 0;

}



**59.** **Write a c++ program to create a class for a rectangle with a constructor and a destructor**

#include <iostream>

class Rectangle {

private:

int length;

int width;

public:

Rectangle(int l, int w) : length(l), width(w) {

std::cout << "Rectangle object created with length " << length << " and width " << width << std::endl;

}

~Rectangle() {

std::cout << "Rectangle object destroyed." << std::endl;

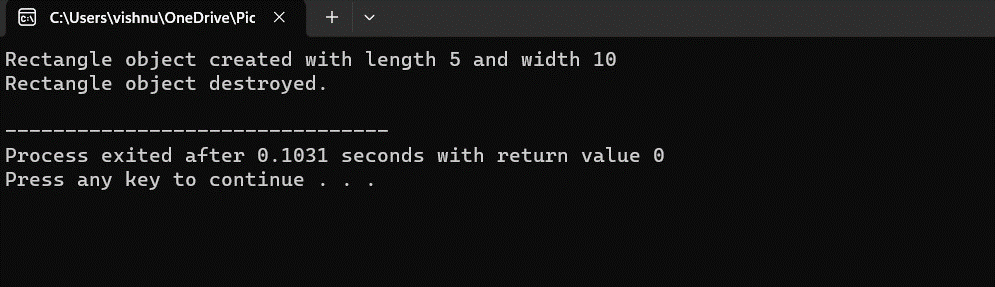
}

};

int main() {

Rectangle myRectangle(5, 10);

return 0;

}

**60.** **Write a c++ program to create a class for a book with a constructor and a destructor.**

#include <iostream>

#include <string>

class Book {

private:

std::string title;

std::string author;

public:

Book(const std::string& t, const std::string& a) : title(t), author(a) {

std::cout << "Book object created: " << title << " by " << author << std::endl;

}

~Book() {

std::cout << "Book object destroyed: " << title << " by " << author << std::endl;

}

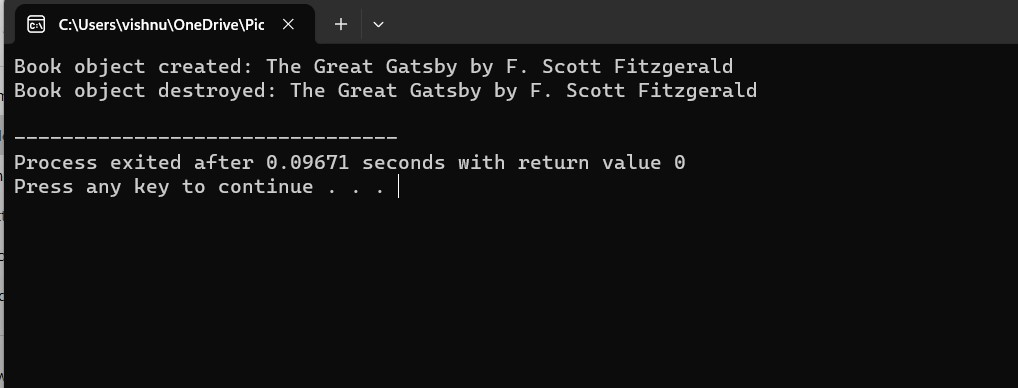
};

int main() {

Book myBook("The Great Gatsby", "F. Scott Fitzgerald");

return 0;

}



**61. Write a c++ program to create a class for student with a constructor and a destructor.**

#include <iostream>

#include <string>

class Student {

private:

std::string name;

int age;

public:

Student(const std::string& n, int a) : name(n), age(a) {

std::cout << "Student object created: " << name << " (Age: " << age << ")" << std::endl;

}

~Student() {

std::cout << "Student object destroyed: " << name << " (Age: " << age << ")" << std::endl;

}

};

int main() {

Student myStudent("Alice", 20);

return 0;

}

