1. WAP to Check Prime Number using loop?

//C++ : check if number is prime

#include<iostream>

using namespace std;

int main(){

int n,i=2;

cin>>n;

while(i<=n/2){

if(n%i==0)

break;

i++;

}

if(i<=n/2)

cout<<"Not Prime"<<endl;

else

cout<<"Prime"<<endl;

return 0;

}

=============================================================

1. WAP to Check if a Number is Positive or Negative

#include<iostream>

using namespace std;

int main ()

{

int num;

cout<< "Enter the number to be checked : ";

cin>>num;

if (num>= 0)

cout<<num<< " is a positive number.";

else

cout<<num<< " is a negative number.";

return 0;

}

==================================================================================

1. WAP to Find Factorial of a number

#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;

}

==================================================================================

1. WAP to Display Fibonacci series

#include <iostream>

using namespace std;

int main() {

int n1=0,n2=1,n3,i,number;

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

cin>>number;

cout<<n1<<" "<<n2<<" ";

for(i=2;i<number;++i)

{

n3=n1+n2;

cout<<n3<<" ";

n1=n2;

n2=n3;

}

return 0;

}

==================================================================================

5. WAP to Display Fibonacci series up to a given

number (instead of terms)

Input: 20

Output:0 + 1 + 1 + 2 + 3 + 5 + 8 + 13

#include<iostream>

using namespace std;

int main()

{

int limit, first=0, second=1, next, num;

cout<<"Enter the limit of Fibonacci series"<<endl;

cin>>num;

for(int p=0;p<num;p++)

{

if (p <= 1)

next = p;

else

{

next = first + second;

first = second;

second = next;

}

cout<<next<<" ";

}

return 0;

}

==================================================================================

1. WAP to Display Uppercased A to Z

void uppercaseAlphabets()

{

for (char c = 'A'; c <= 'Z'; ++c)

cout<< c << " ";

cout<<endl;

}

int main()

{

cout<< "Uppercase Alphabets" <<endl;

uppercaseAlphabets(ch);

cout<< "Lowercase Alphabets " <<endl;

lowercaseAlphabets(ch);

return 0;

}

===========================================================================

1. WAP to Swap two numbers using temporary variable

#include <iostream>

using namespace std;

int main()

{

int a = 5, b = 10, temp;

cout<<"Before swapping."<<endl;

cout<<"a ="<<a<<",b ="<<b<<endl;

temp = a;

a = b;

b = temp;

cout<< "\nAfter swapping." <<endl;

cout<< "a = "<< a << ", b = "<< b <<endl;

return 0;

}

===========================================================================

8. WAP to Swap two numbers without using temporary

Variable

#include <iostream>

using namespace std;

int main()

{

int a=5, b=10;

cout<<"Before swap a= "<<a<<" b= "<<b<<endl;

a=a\*b;

b=a/b;

a=a/b;

cout<<"After swap a= "<<a<<" b= "<<b<<endl;

return 0;

}

===============================================================================

9. WAP to Check whether an alphabet is vowel or

Consonant

#include <iostream>

using namespace std;

int main()

{

char c;

int isLowercaseVowel, isUppercaseVowel;

cout<< "Enter an alphabet: ";

cin>> c;

isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');

if (isLowercaseVowel || isUppercaseVowel)

cout<< c << " is a vowel.";

else

cout<< c << " is a consonant.";

return 0;

}

==================================================================================

10. WAP to Find the largest number among the given

Numbers

#include <iostream>

using namespace std;

int main()

{

int a=3,b=4,c=10;

if(a>b && a>c){

cout<<"is greatest among three numbers"<<a;

}

else if(b>a && b>c){

cout<<"is greatest among three numbers"<<b;

}

else if(c>a && c>b){

cout<< "is greatest among three numbers"<<c;

}

return 0;

}

==================================================================================

11. WAP to Reverse a Number

#include <iostream>

using namespace std;

int main() {

int n, reversedNumber = 0, remainder;

cout<< "Enter an integer:";

cin>> n;

while(n != 0) {

remainder = n%10;

reversedNumber = reversedNumber\*10 + remainder;

n /= 10;

}

cout<< "Reversed Number ="<<reversedNumber;

return 0;

}

==================================================================================

12. WAP to Program to Check Palindrome

#include <iostream>

using namespace std;

int main()

{

int n,r,sum=0,temp;

cout<<"Enter the Number=";

cin>>n;

temp=n;

while(n>0)

{

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

cout<<"Number is Palindrome.";

else

cout<<"Number is not Palindrome.";

return 0;

}

==================================================================================

13. WAP to Program to Check Prime Number

==================================================================================

14. WAP to Display Prime Numbers Between two

Intervals

#include <iostream>

using namespace std;

int main() {

int low, high, i;

bool isPrime = true;

cout<< "Enter two numbers (intervals): ";

cin>> low >> high;

cout<< "\nPrime numbers between " << low << " and " << high << " are: " <<endl;

while (low < high) {

isPrime = true;

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

isPrime = false;

}

else {

for (i = 2; i<= low / 2; ++i) {

if (low % i == 0) {

isPrime = false;

break;

}

}

}

if (isPrime)

cout<< low << " ";

++low;

}

return 0;

}

==================================================================================

15. WAP to Check Armstrong Number for 3 digit

Number

#include <iostream>

using namespace std;

int main() {

int num, originalNum, remainder, result = 0;

cout<< "Enter a three-digit integer: ";

cin>>num;

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

originalNum /= 10;

}

if (result == num)

cout<<num<< " is an Armstrong number.";

else

cout<<num<< " is not an Armstrong number.";

return 0;

}