**26 – 35 questions**

// Q26. generate first n prime numbers (for-while)

#include<iostream>

using namespace std;

int main(){

    int n;

    cout<<"Enter number of prime numbers: ";

    cin>>n;

    int count=0,num=2;

    while(count<n){

        bool isPrime = true;

        for(int i=2;i<=n;i++){

            if (num % i == 0) {

                isPrime=false;

                break;

            }

        }

        if (isPrime) {

            cout<<num<<" ";

            count++;

        }

        num++;

    }

    cout<<endl;

    return 0;

}

// Q27. all armstrong no's given range using nested loop

#include<iostream>

#include<math.h>

using namespace std;

int main(){

    int start,end;

    cout<<"Enter range (start and end): ";

    cin>>start>>end;

    for (int i = start; i <= end; i++) {

    int num = i, temp = num, sum = 0, n = 0;

    int countdigit = num;

    while (countdigit > 0) {

        countdigit /= 10;

        n++;

    }

    temp = num;

    while (temp > 0) {

        int digit = temp % 10;

        sum += pow(digit, n);

        temp /= 10;

    }

    if (sum == num) {

        cout << num << " ";

    }

}

cout << endl;

return 0;

}

// Q28. simulate a number guessing game where user has limited attempts to guess a randomly generated number.

// Provide feedback(too high, too low) and terminate the game when no. is guessed or attempts are exhausted

#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

    int secretNumber = (rand() % 100) + 1;

    int attempts = 7;

    int guess;

    cout << "Guess the number (between 1 and 100). You have " << attempts << " attempts." << endl;

    while (attempts > 0) {

        cout << "Enter your guess: ";

        cin >> guess;

        if (guess == secretNumber) {

            cout << "Congratulations! You guessed the correct number." << endl;

            return 0;

        } else if (guess < secretNumber) {

            cout << "Too low! Try again." << endl;

        } else {

            cout << "Too high! Try again." << endl;

        }

        attempts--;

        if (attempts > 0) {

            cout << "Attempts remaining: " << attempts << endl;

        } else {

            cout << "Game over! The correct number was " << secretNumber << "." << endl;

        }

    }

    return 0;

}

// Q29. first number greater than 50 that is divisible by 7

#include<iostream>

using namespace std;

int main(){

    for(int n=51; ;n++){

        if(n % 7 == 0){

            cout<<"The first number greater than 50 that is divisible by 7 is : "<<n;

            break;

        }

    }

    return 0;

}

// Q30. print all number between 1 500, skip numbers divisible by both 3 and 7. stop printing numbers if sum of printed numbers exceeds 1000

#include<iostream>

using namespace std;

int main(){

    int sum=0;

    for(int i=0;i<=500;i++){

        if((i % 3 == 0) && (i % 7 == 0)){

            continue;;

        }

        if (sum + i > 1000) {

            break;

        }

        cout << i << " ";

        sum += i;

    }

    return 0;

}

// Q31. reverse digits of given number and terminate early if reversed number becomes a palindrome

#include<iostream>

using namespace std;

int main(){

    int num;

    cout<<"Enter a number: ";

    cin>>num;

    int reversedNum=0,originalNum=num;

    while (num>0) {

        int digit = num % 10;

        reversedNum = reversedNum \* 10 + digit;

        num = num / 10;

        if (reversedNum==originalNum){

            cout<<"The given number is a palindrome."<<endl;

            break;

        }

    }

    if (reversedNum != originalNum) {

        cout<<"Reversed Number: "<<reversedNum<<endl;

    }

    return 0;

}

// Q32. takes integer array from user and finds the second largest element

#include<iostream>

using namespace std;

int main(){

    int n;

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

    cin>>n;

    int arr[n];

    for(int i=0;i<n;i++){

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

        cin>>arr[i];

    }

    int largest = arr[0];

    int secondLargest = arr[1];

    for (int i=2;i<n;i++) {

        if (arr[i]>largest) {

            secondLargest=largest;

            largest=arr[i];

        } else if (arr[i]>secondLargest && arr[i]!=largest) {

            secondLargest=arr[i];

        }

    }

    if (secondLargest == largest)

        cout<<"There is no second largest element"<<endl;

    else

        cout<<"Second Largest Element: "<<secondLargest<<endl;

    return 0;

}

// Q33. determine whether a given floating point number can be represented exactly in binary format and explain why

// binary representation of floating point numbers

#include <iostream>

#include <cmath>

using namespace std;

bool isExactlyRepresentable(double num) {

    while (num != floor(num)) {

        num \*= 2;

        if (num > floor(num)) {

            return false;

        }

    }

    return true;

}

int main() {

    double number;

    cout << "Enter a floating-point number: ";

    cin >> number;

    if (isExactlyRepresentable(number)) {

        cout << number << " can be represented exactly in binary." << endl;

    } else {

        cout << number << " cannot be represented exactly in binary." << endl;

    }

    return 0;

}

// Q34. takes input for a 2-D array and prints array in formatted table

#include<iostream>

using namespace std;

int main(){

    int rows,col;

    cout<<"Enter number of rows and column: ";

    cin>>rows>>col;

    int arr[rows][col];

    cout<<"Entering the elements of 2-D array: "<<endl;

    for(int i=0;i<rows;i++){

        for(int j=0;j<col;j++){

            cout<<"Enter element "<<"["<<i<<"] "<<"["<<j<<"] ";

            cin>>arr[i][j];

        }

    }

    for(int i=0;i<rows;i++){

        for(int j=0;j<col;j++){

                cout<<arr[i][j]<<"\t";

        }

        cout<<endl;

    }

    return 0;

}

// Q35. calculate LCM and GCD of two integers

#include<iostream>

using namespace std;

int main(){

    int num1, num2;

    cout << "Enter two numbers: ";

    cin >> num1 >> num2;

    int originalNum1 = num1;

    int originalNum2 = num2;

    while (num2 != 0) {

        int temp = num2;

        num2 = num1 % num2;

        num1 = temp;

    }

    int gcd = num1;

    int lcm = (originalNum1 \* originalNum2) / gcd;

    cout<<"GCD of "<< originalNum1<<" and "<<originalNum2<<" is: "<<gcd<<endl;

    cout<<"LCM of "<< originalNum1<<" and "<<originalNum2<<" is: "<<lcm<<endl;

    return 0;

}