

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

Loop (part one)

Question 1:

Write a program that asks the user to enter a positive integer and then prints all the numbers from 1 to that integer.

1. Solution using a for loop:

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;

    cout << "Using a for loop:" << endl;
    for (int i = 1; i <= num; i++) {
        cout << i << " ";
    }
    cout << endl;

    return 0;
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

2. Solution using a while loop:

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;

    cout << "Using a while loop:" << endl;
    int i = 1;
    while (i <= num) {
        cout << i << " ";
        i++;
    }
    cout << endl;

    return 0;
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

3. Solution using a do-while loop:

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int num;  
    cout << "Enter a positive integer: ";  
    cin >> num;  
  
    cout << "Using a do-while loop:" << endl;  
    int i = 1;  
    do {  
        cout << i << " ";  
        i++;  
    } while (i <= num);  
    cout << endl;  
  
    return 0;  
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

Question 2:

Write a program that prompts the user to enter a number and then prints the multiplication table for that number up to 10.

1. Solution using a for loop:

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    cout << "Using a for loop:" << endl;
    for (int i = 1; i <= 10; i++) {
        cout << num << " * " << i << " = " << num * i << endl;
    }

    return 0;
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

2. Solution using a while loop:

```
#include <iostream>
using namespace std;

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    cout << "Using a while loop:" << endl;
    int i = 1;
    while (i <= 10) {
        cout << num << " * " << i << " = " << num * i << endl;
        i++;
    }

    return 0;
}
```

3. Solution using a do-while loop:

```
#include <iostream>
using namespace std;
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
int main() {  
    int num;  
    cout << "Enter a number: ";  
    cin >> num;  
  
    cout << "Using a do-while loop:" << endl;  
    int i = 1;  
    do {  
        cout << num << " * " << i << " = " << num * i << endl;  
        i++;  
    } while (i <= 10);  
  
    return 0;  
}
```

4. Solution using a goto statement:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int num;  
    cout << "Enter a number: ";  
    cin >> num;  
  
    cout << "Using goto:" << endl;  
    int i = 1;
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

start:

```
cout << num << " * " << i << " = " << num * i << endl;
```

```
i++;
```

```
if (i <= 10)
```

```
    goto start;
```

```
return 0;
```

```
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

1. Write a program in C++ to find the first 10 natural numbers.

Sample output:

The natural numbers are:

1 2 3 4 5 6 7 8 9 10

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    cout << "The natural numbers are:" << endl;  
  
    for (int i = 1; i <= 10; i++) {  
        cout << i << " ";  
    }  
  
    return 0;  
}
```


STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

2. Write a program in C++ to find the sum of the first 10 natural numbers.

Sample Output:

Find the first 10 natural numbers:

The natural numbers are:

1 2 3 4 5 6 7 8 9 10

The sum of first 10 natural numbers: 55

solution :

```
#include <iostream>
```

```
int main() {  
    int sum = 0;
```

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

```
    std::cout << "The sum of the first 10 natural numbers is: " << sum <  
< std::endl;
```

```
    return 0;  
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

3. Write a program in C++ to display n terms of natural numbers and their sum.

Sample Output:

Input a number of terms: 7

The natural numbers upto 7th terms are:

1 2 3 4 5 6 7

The sum of the natural numbers is: 28

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```
    int sum = 0;
```

```
    cout << "Input a number of terms: ";
```

```
    cin >> n;
```

```
    cout << "The natural numbers up to the " << n << "th terms are:"  
<< endl;
```

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

```
        cout << i << " ";
```

```
        sum += i;
```

```
    }
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
cout << "\n\nThe sum of the natural numbers is: " << sum << endl;
```

```
return 0;
```

```
}
```

هذا البرنامج يقوم بنفس الوظيفة السابقة، ولكنه يستخدم **using namespace std** لتبسيط استخدام الدوال المعرفة في مساحة الأسماء **std**. بالتالي، يمكننا استخدام **cout** و **cin** بدون كتابة **std::** قبل كل استخدام. عند تشغيل البرنامج وإدخال القيمة **7** كعدد العناصر، ستكون النتيجة كما هو متوقع:

The natural numbers up to the 7th terms are:

1 2 3 4 5 6 7

The sum of the natural numbers is: 28

4. Write a program in C++ to check whether a number is prime or not.

Sample Output:

Input a number to check prime or not: 13

The entered number is a prime number.

Solution :

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int num;
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
int isPrime = 1;
```

```
cout << "Input a number to check prime or not: ";  
cin >> num;
```

```
for (int i = 2; i <= num / 2; i++) {  
    if (num % i == 0) {  
        isPrime = 0;  
        break;  
    }  
}
```

```
if (isPrime == 1 && num > 1) {  
    cout << "The entered number is a prime number." << endl;  
} else {  
    cout << "The entered number is not a prime number." << endl;  
}
```

```
return 0;
```

```
}
```

ر **isPrime** من نوع **bool** إلى **int** وتم تعيينه إلى **1** للتمثيل البديل للقيمة **true** و **0** للتمثيل البديل للقيمة **false**.

في حلقة **for**، إذا تم العثور على عامل يقسم **num** بدون بقية، يتم تعيين **isPrime** إلى **0** والخروج من الحلقة باستخدام **break**.

بعد الحلقة، يتم استخدام التعبير **isPrime == 1** في الشرط في الجملة الشرطية **if** للتحقق مما إذا كان **isPrime** يساوي **1** وأن العدد **num** أكبر من **1**. إذا كانت كلا الشروط صحيحة، يتم طباعة **"The**

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

"The entered number is **not a prime number.**"
وإلا يتم طباعة **entered number is a prime number.**

عند تشغيل البرنامج وإدخال القيمة 13 كرقم للتحقق

6. Write a program in C++ to find a prime number within a range.

Input number for starting range: 1

Input number for ending range: 100

The prime numbers between 1 and 100 are:

**2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97**

The total number of prime numbers between 1 to 100 is: 25

Solution :

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int startRange, endRange;
```

```
    int count = 0;
```

```
    cout << "Enter the starting range number: ";
```

```
    cin >> startRange;
```

```
    cout << "Enter the ending range number: ";
```

```
    cin >> endRange;
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
cout << "The prime numbers between " << startRange << " and " <<
endRange << " are:" << endl;
```

```
for (int i = startRange; i <= endRange; i++) {
    int isPrime = 1;
```

```
    for (int j = 2; j <= i / 2; j++) {
        if (i % j == 0) {
            isPrime = 0;
            break;
        }
    }
}
```

```
    if (isPrime == 1) {
        cout << i << " ";
        count++;
    }
}
```

```
cout << "\n\nThe total number of prime numbers between " << startRa
nge << " and " << endRange << " is: " << count << endl;
```

```
return 0;
}
```

في هذا البرنامج، نقوم بتعريف المتغيرات **startRange** و **endRange** لتخزين إدخال المستخدم للنطاق البدني والنهائي. نقوم أيضاً بتعريف المتغير **count** لتتبع عدد الأعداد الأولية التي تم العثور عليها.

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

يتم طلب إدخال المستخدم للنطاق البدني والنهائي باستخدام **cin.** و **cout** باستخدام حلقة **for**، نقوم بتكرار العدادات في النطاق المحدد. لكل عدد، نقوم بتعريف المتغير **isPrime** من نوع **int** ونعطيه القيمة **1** للفرض أن العدد هو عدد أولي. داخل الحلقة المدمجة، نتحقق مما إذا كان العدد قابلاً للقسمة على أي عدد بين **2** ونصف العدد نفسه بدون بقية. إذا كان العدد قابلاً للقسمة، فإننا نعيد المتغير **isPrime** قيمة **0** وننقطع عن الحلقة. بعد الحلقة المدمجة، نتحقق مما إذا كان قيمة المتغير **isPrime** ما زالت **1**. إذا كانت كذلك، يعني أن العدد هو عدد أولي، ونقوم بطباعته وزيادة قيمة المتغير **count.** أخيراً، نقوم بإخراج إجمالي عدد الأعداد الأولية التي تم العثور عليها بين النطاق المحدد باستخدام **cout.** عند تشغيل هذا البرنامج وتوفير الإدخال **1** للنطاق البدني و **100** للنطاق النهائي، سيتم إخراج نفس النتيجة كما سبق:

الأعداد الأولية بين **1** و **100** هي:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

إجمالي عدد الأعداد الأولية بين **1** و **100** هو **25** :

7. Write a program in C++ to find the factorial of a number.

Sample output:

Input a number to find the factorial: 5

The factorial of the given number is: 120

Solution :

```
#include <iostream>  
using namespace std;
```

```
int main() {
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
int num;
int factorial = 1;

cout << "Input a number to find the factorial: ";
cin >> num;

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

cout << "The factorial of the given number is: " << factorial << endl;
return 0;
}
```

في هذا البرنامج، نقوم بتعريف المتغيرات **num** و **factorial**. نطلب من المستخدم إدخال عدد لحساب عامله باستخدام **cout** ونقوم بقبول الإدخال باستخدام **cin**. ثم نستخدم حلقة **for** للتكرار من 1 إلى العدد المعطى ونقوم بحساب عامل العدد عن طريق ضرب المتغير **factorial** بقيمة المتغير **i** في كل تكرار. أخيرًا، نقوم بطباعة عامل العدد المعطى باستخدام **cout**. عند تشغيل البرنامج وإدخال العدد 5، سيتم طباعة الناتج كالتالي:

Copy

أدخل عددًا لحساب عامله: 5

عامل العدد المعطى هو: 120

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

8. Write a program in C++ to find the sum of the series $1 + 1/2^2 + 1/3^3 + \dots + 1/n^n$.

Sample Output:

Input the value for nth term: 5

$$1/1^1 = 1$$

$$1/2^2 = 0.25$$

$$1/3^3 = 0.037037$$

$$1/4^4 = 0.00390625$$

$$1/5^5 = 0.00032$$

The sum of the above series is: 1.29126

[Click me to see the sample solution](#)

9. Write a program in C++ to calculate the sum of the series $(1*1) + (2*2) + (3*3) + (4*4) + (5*5) + \dots + (n*n)$.

Sample Output:

Input the value for nth term: 5

$$1*1 = 1$$

$$2*2 = 4$$

$$3*3 = 9$$

$$4*4 = 16$$

$$5*5 = 25$$

The sum of the above series is: 55

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

Solution :

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int n;  
    int sum = 0;
```

```
    cout << "Input the value for nth term: ";  
    cin >> n;
```

```
    for (int i = 1; i <= n; i++) {  
        int term = i * i;  
        cout << i << "*" << i << " = " << term << endl;  
        sum += term;  
    }
```

```
    cout << "The sum of the above series is: " << sum << endl;
```

```
    return 0;
```

```
}
```

في هذا البرنامج، نقوم بتعريف المتغيرات **sum** و **n** ونطلب من المستخدم إدخال قيمة العنصر النهائي **n** للسلسلة باستخدام **cin** ونقوم بقبول الإدخال باستخدام **cout**.

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

ثم نستخدم حلقة **for** للتكرار من 1 إلى القيمة المعطاة لـ **n**. في كل تكرار، نقوم بحساب مربع الرقم الحالي ونخزنه في المتغير **term**، ونقوم بطباعة العنصر بالصيغة "**i*i = term**" باستخدام **cout**. نقوم أيضاً بتحديث المتغير **sum** عن طريق إضافة العنصر الحالي إليه. أخيراً، نقوم بطباعة مجموع السلسلة باستخدام **cout**. عند تشغيل البرنامج وإدخال القيمة 5 للعنصر النهائي، سيتم طباعة الناتج كالتالي:

apache

Copy

أدخل قيمة العنصر النهائي 5: (n):

$$1 * 1 = 1$$

$$2 * 2 = 4$$

$$3 * 3 = 9$$

$$4 * 4 = 16$$

$$5 * 5 = 25$$

مجموع السلسلة أعلاه هو 55 :

10. Write a program in C++ to calculate the series (1) + (1+2) + (1+2+3) + (1+2+3+4) + ... + (1+2+3+4+...+n).

Sample Output:

Input the value for nth term: 5

$$1 = 1$$

$$1+2 = 3$$

$$1+2+3 = 6$$

$$1+2+3+4 = 10$$

$$1+2+3+4+5 = 15$$

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

The sum of the above series is: 35

Solution:

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int n;  
    int sum = 0;  
  
    cout << "Input the value for nth term: ";  
    cin >> n;  
  
    for (int i = 1; i <= n; i++) {  
        int term = 0;  
        for (int j = 1; j <= i; j++) {  
            term += j;  
        }  
        cout << term;  
        sum += term;  
  
        if (i != n) {  
            cout << " + ";  
        }  
    }  
}
```

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
cout << endl << "The sum of the above series is: " << sum << endl;
```

```
return 0;
```

```
}
```

في هذا البرنامج، نقوم بتعريف المتغيرات **sum** و **n**. نطلب من المستخدم إدخال القيمة المطلوبة للعنصر النهائي **n** للسلسلة باستخدام **cout** ونقوم بقبول الإدخال باستخدام **cin**. نستخدم حلقتي **for** متداخلتين. الحلقة الخارجية تكرر من **1** إلى القيمة المعطاة لـ **n** وتمثل كل عنصر في السلسلة. لكل عنصر، نقوم بتهيئة المتغير **term** للقيمة **0**. الحلقة الداخلية تكرر من **1** إلى القيمة الحالية للمتغير الخارجي **i** وتمثل مجموع الأعداد من **1** إلى **i**. نقوم بإضافة كل عدد إلى المتغير **term**. نقوم بطباعة العنصر الحالي باستخدام **cout** ونقوم بتحديث المتغير **sum** عن طريق إضافة العنصر الحالي إليه. إذا كان العنصر الحالي ليس العنصر الأخير (**i != n**)، نقوم بطباعة " + " لفصل العناصر ببعضها. أخيرًا، نقوم بطباعة سطر جديد ومجموع السلسلة باستخدام **cout**. عند تشغيل البرنامج وإدخال القيمة **5** للعنصر النهائي، سيتم طباعة الناتج كالتالي:

أدخل قيمة العنصر النهائي **5 (n):**

$$1 = 1$$
$$1 + 2 = 3$$
$$1 + 2 + 3 = 6$$
$$1 + 2 + 3 + 4 = 10$$
$$1 + 2 + 3 + 4 + 5 = 15$$

مجموع السلسلة أعلاه هو **35** :

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

11. Write a program in C++ to print a square pattern with the # character.

Sample Output:

Print a pattern like square with # character:

Input the number of characters for a side: 4

```
# # # #  
# # # #  
# # # #  
# # # #
```

Solution :

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int side;
```

```
    cout << "Input the number of characters for a side: ";  
    cin >> side;
```

```
    cout << "Print a pattern like square with # character:" << endl;  
    cout << "-----" << endl;
```

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

STRUCTURED PROGRAMMING

C++ LANGUAGE

By
Noor Muwafak

```
    cout << "# ";  
}  
cout << endl;  
}  
  
return 0;  
}
```