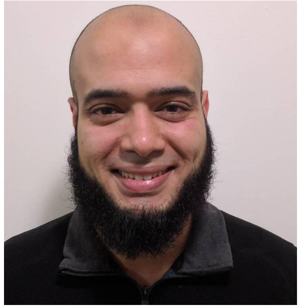
Programming4kids

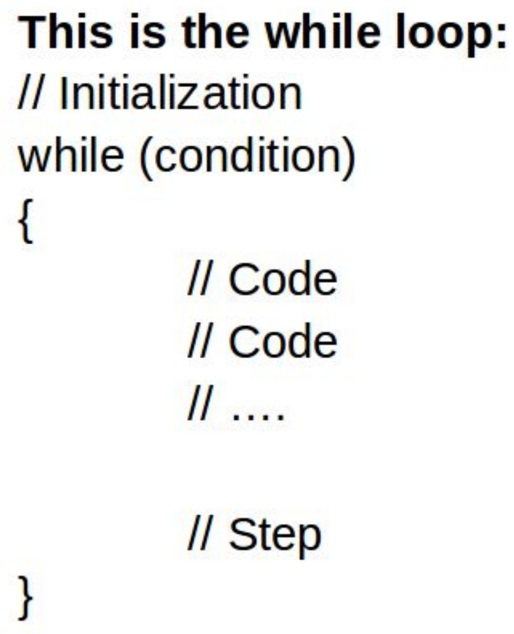
For Loops

Mostafa Saad Ibrahim
Computer Vision Researcher @ Huawei Canada
PhD - Simon Fraser University
Bachelor / Msc - FCI Cairo University

Ex-(Software Engineer / Teaching Assistant)

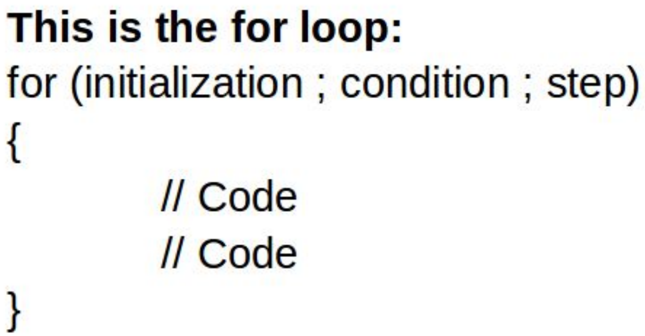


While ⇒ For



This is the while loop:

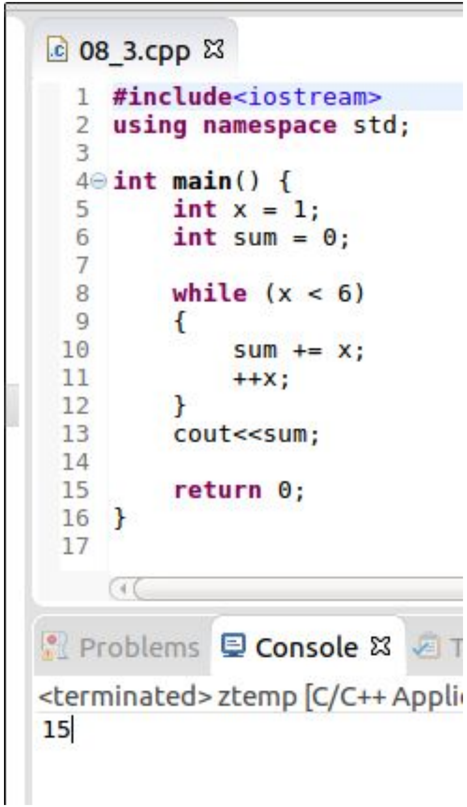
```
// Initialization
while (condition)
{
        // Code
        // Code
        // ....

        // Step
}
```



This is the for loop:

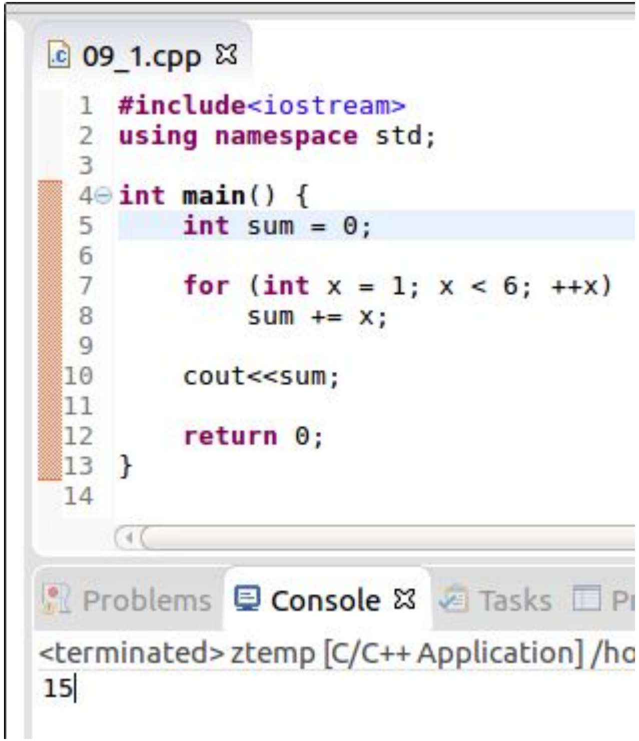
```
for (initialization ; condition ; step)
{
        // Code
        // Code
}
```

While ⇒ For



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

int main() {
    int x = 1;
    int sum = 0;

    while (x < 6)
    {
        sum += x;
        ++x;
    }
    cout<<sum;

    return 0;
}

```

```
<terminated> ztemp [C/C++ Appli
15
```



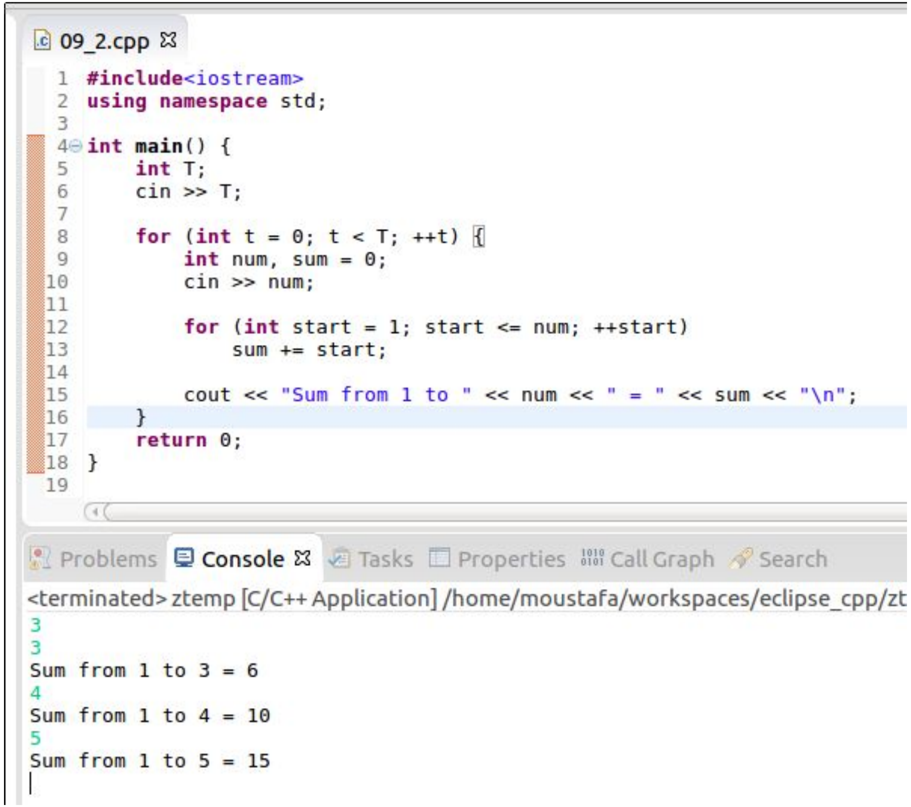
```cpp
#include<iostream>
using namespace std;

int main() {
    int sum = 0;

    for (int x = 1; x < 6; ++x)
        sum += x;

    cout<<sum;

    return 0;
}

```

```
<terminated> ztemp [C/C++ Application] /ho
15
```

Order

- Initialization
- Condition
- Body
- Step
- Condition
- Body
- Step
- Condition
- Body
- Step
- ...
- Condition
  - END

Nested for loops



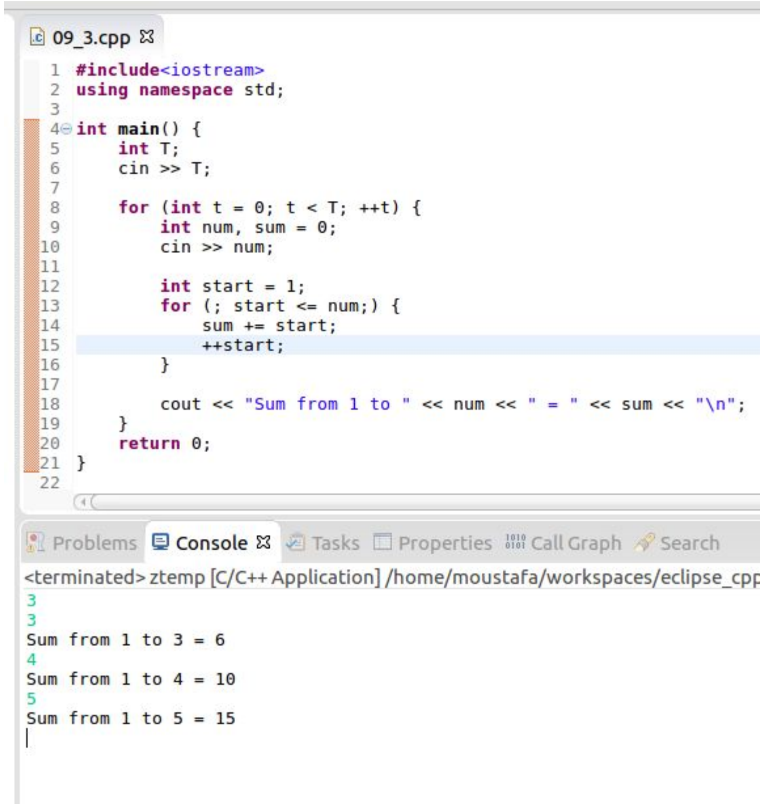
```cpp
#include<iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; ++t) {
        int num, sum = 0;
        cin >> num;

        for (int start = 1; start <= num; ++start)
            sum += start;

        cout << "Sum from 1 to " << num << " = " << sum << "\n";
    }
    return 0;
}
```

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cpp/zt

```
3
3
Sum from 1 to 3 = 6
4
Sum from 1 to 4 = 10
5
Sum from 1 to 5 = 15
```

- Read T test cases
  - Read integer N
  - Print sum from 1 to N

Loop empty parts

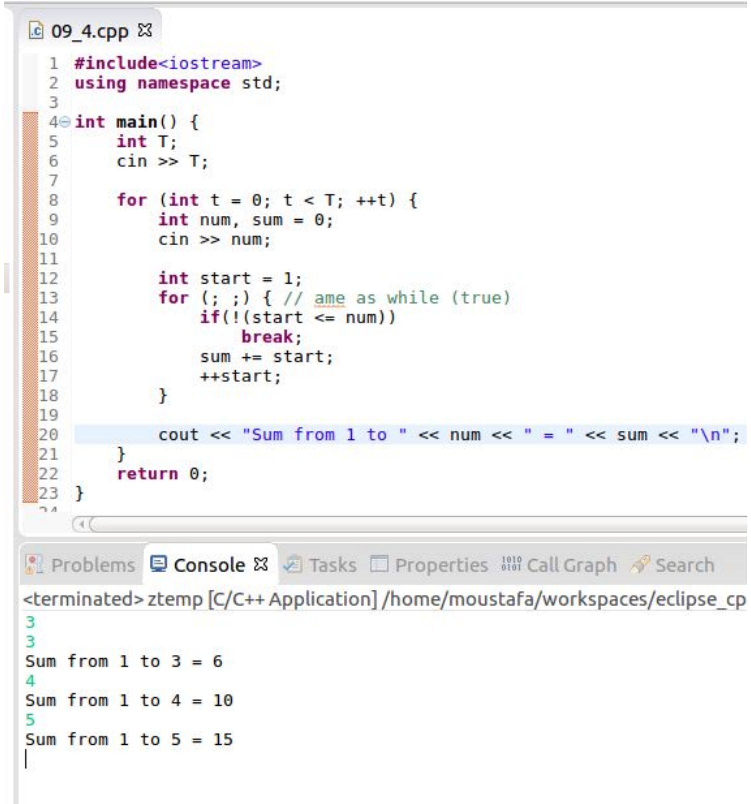


```
#include<iostream>
using namespace std;

int main() {
	int T;
	cin >> T;

	for (int t = 0; t < T; ++t) {
		int num, sum = 0;
		cin >> num;

		int start = 1;
		for (; start <= num;) {
			sum += start;
			++start;
		}

		cout << "Sum from 1 to " << num << " = " << sum << "\n";
	}
	return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cpp
3
3
Sum from 1 to 3 = 6
4
Sum from 1 to 4 = 10
5
Sum from 1 to 5 = 15
```

- We can remove 2 elements of the for loops, now exact while

Loop empty parts

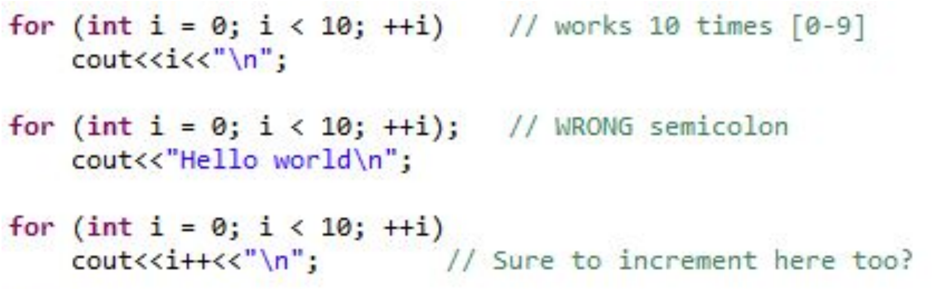


```cpp
#include<iostream>
using namespace std;

int main() {
    int T;
    cin >> T;

    for (int t = 0; t < T; ++t) {
        int num, sum = 0;
        cin >> num;

        int start = 1;
        for (; ;) { // ame as while (true)
            if(!(start <= num))
                break;
            sum += start;
            ++start;
        }

        cout << "Sum from 1 to " << num << " = " << sum << "\n";
    }
    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cp
3
3
Sum from 1 to 3 = 6
4
Sum from 1 to 4 = 10
5
Sum from 1 to 5 = 15
```

- We can even remove the 3 elements!

Common Errors

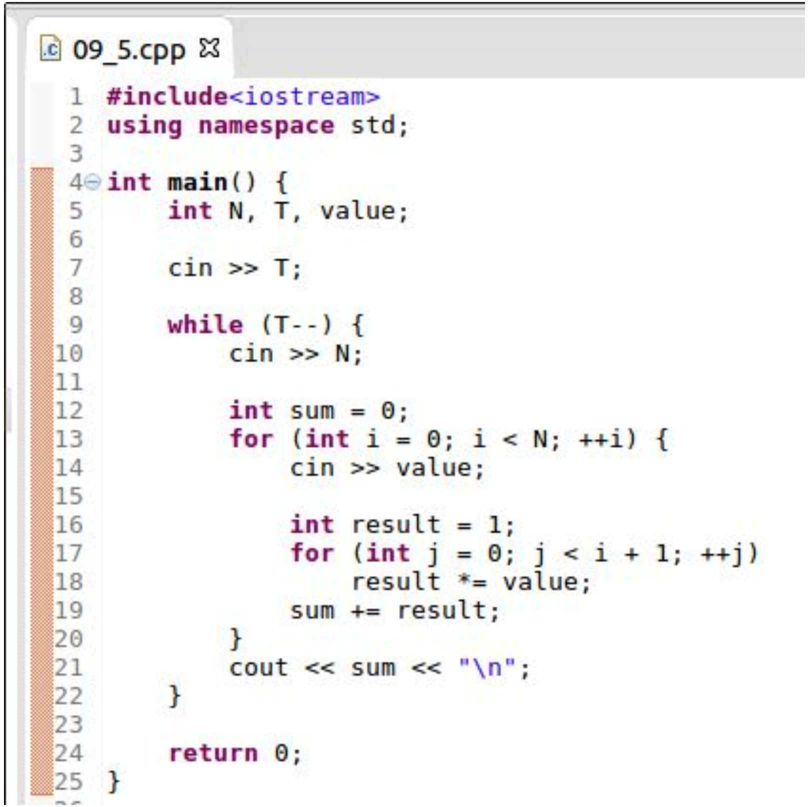


```
for (int i = 0; i < 10; ++i)    // works 10 times [0-9]
    cout<<i<<"\n";

for (int i = 0; i < 10; ++i);   // WRONG semicolon
    cout<<"Hello world\n";

for (int i = 0; i < 10; ++i)
    cout<<i++<<"\n";        // Sure to increment here too?
```

Practice: Special Sum

- Read T for number of test cases. For each test case read integer N: number of numbers. Then read N numbers a, b, c, ….. and compute the **sum** of:
    - (a, b*b, c*c*c, d*d*d*d, e*e*e*e*e……)
    - That is the k-th number is repeated k times
- Input:
    - 2
    - 3   5 7 2
    - 4 1 2 3 4
- Output
    - 62   [as (5 + 7*7 + 2*2*2) = 62]
    - 288   [as (1+2*2+3*3*3+4*4*4*4) = 288]
- Stop video and code using for loop

Practice: Special Sum

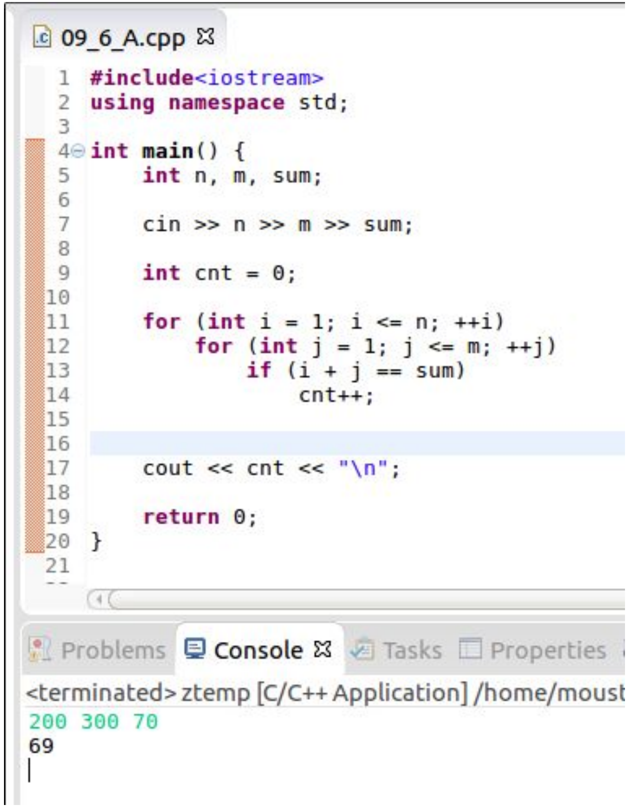


```
#include<iostream>
using namespace std;

int main() {
    int N, T, value;

    cin >> T;

    while (T--) {
        cin >> N;

        int sum = 0;
        for (int i = 0; i < N; ++i) {
            cin >> value;

            int result = 1;
            for (int j = 0; j < i + 1; ++j)
                result *= value;
            sum += result;
        }
        cout << sum << "\n";
    }

    return 0;
}
```

- Let’s rewrite the previous code
- We keep the while as more convenient
    - While (T--) is shortcut for While (T-- != 0)
- Replace as 2 whiles with 2 fors

Practice: Pair of numbers

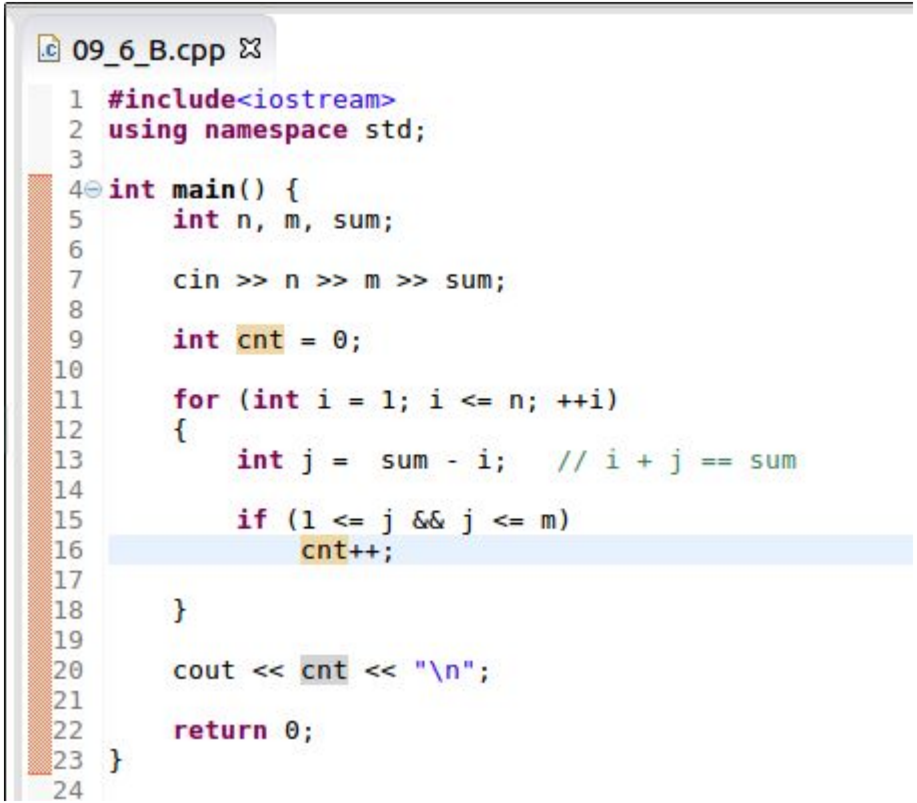
- Read N, M, SUM. Find all pairs that has A + B = SUM where
  - 1 <= A <= N
  - 1 <= B <= M
- Stop video and code
- Try input
  - 1000000 1000000 1000000
  - How many steps the code do?



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

int main() {
    int n, m, sum;

    cin >> n >> m >> sum;

    int cnt = 0;

    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= m; ++j)
            if (i + j == sum)
                cnt++;


    cout << cnt << "\n";

    return 0;
}

```

```
<terminated> ztemp [C/C++ Application] /home/moust
200 300 70
69
```

Practice: Pair of numbers - FASTER

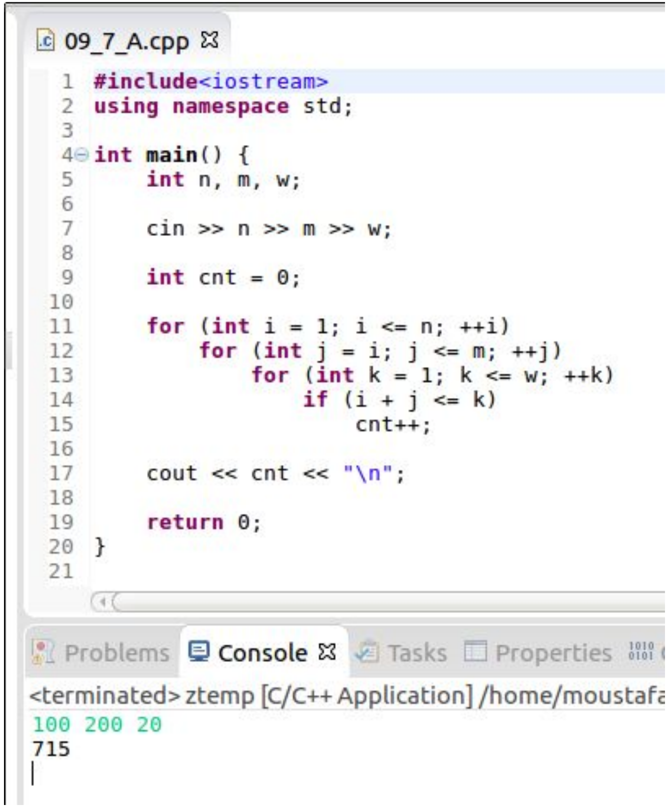


```cpp
#include<iostream>
using namespace std;

int main() {
    int n, m, sum;

    cin >> n >> m >> sum;

    int cnt = 0;

    for (int i = 1; i <= n; ++i)
    {
        int j =  sum - i;   // i + j == sum

        if (1 <= j && j <= m)
            cnt++;

    }

    cout << cnt << "\n";

    return 0;
}
```

- Second loop was useless as only maximum 1 j will have value that matches sum
- With simple math, we can know the possible value of j
  - Then verify its range
- This code takes like 3n steps
  - So for 1000000, just 3-4 milion

Practice: Triples of numbers

- Read N, M, W. Find all triples that has A + B <= C where
    - 1 <= A <= N
    - A <= B <= M
    - 1 <= C <= W
- Stop video and code

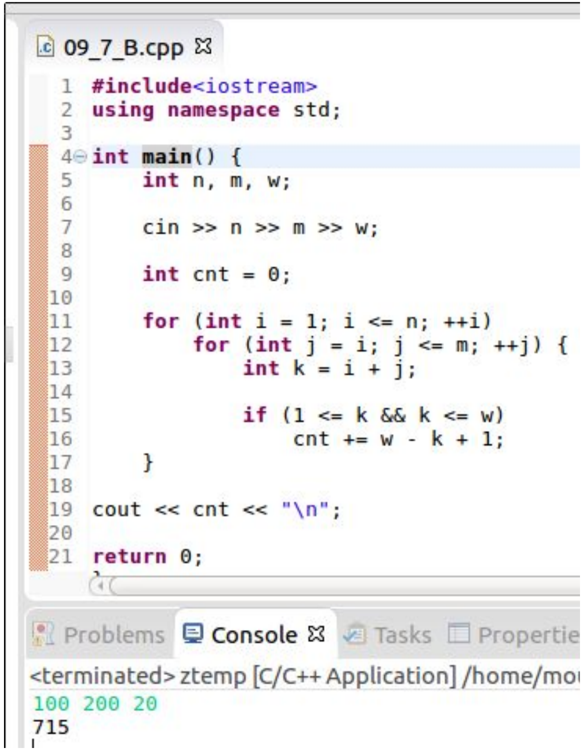


```
#include<iostream>
using namespace std;

int main() {
	int n, m, w;

	cin >> n >> m >> w;

	int cnt = 0;

	for (int i = 1; i <= n; ++i)
		for (int j = i; j <= m; ++j)
			for (int k = 1; k <= w; ++k)
				if (i + j <= k)
					cnt++;

	cout << cnt << "\n";

	return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa
100 200 20
715
```

Practice: Triples of numbers - FASTER

- We can use the same trick
- Remove the very inner loop
- But this is good for i+j == k not i+j <= k ??
- Simply, all ks tell the last k=w are valid
  - So add w-k+1
  - E.g. if k = 4, w = 7
    - Then 4, 5, 6, 7 are correct values

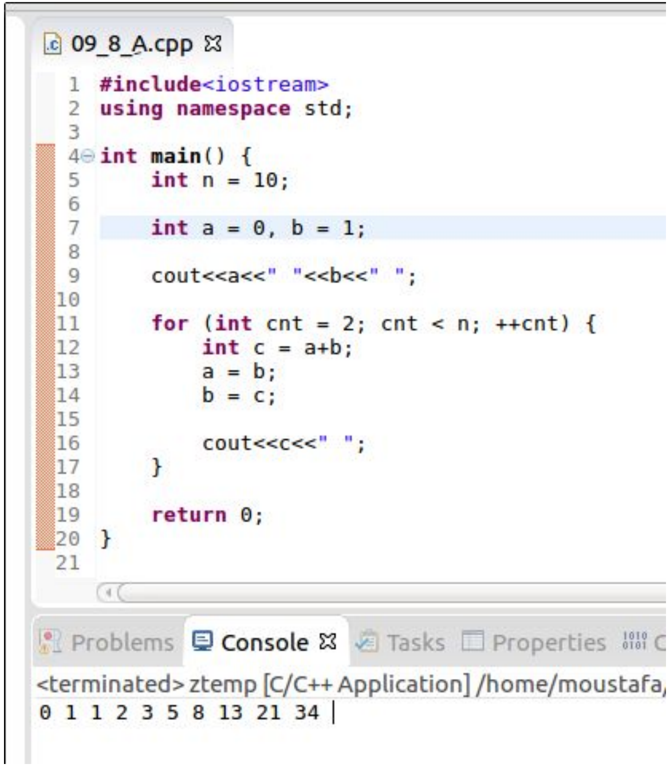


```cpp
#include<iostream>
using namespace std;

int main() {
    int n, m, w;

    cin >> n >> m >> w;

    int cnt = 0;

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

            if (1 <= k && k <= w)
                cnt += w - k + 1;
    }

cout << cnt << "\n";

return 0;
```

```
<terminated> ztemp [C/C++ Application] /home/mo
100 200 20
715
```

Fibonacci Sequence

- A popular math sequence
  - First 2 numbers: 0 1
  - Then each number is sum of last 2 numbers:
    - 0 1 1 2 3 5 8 13 21 34
      - E.g. 13 = 5 + 8
      - E.g. 34 = 13+21
- Write a code!

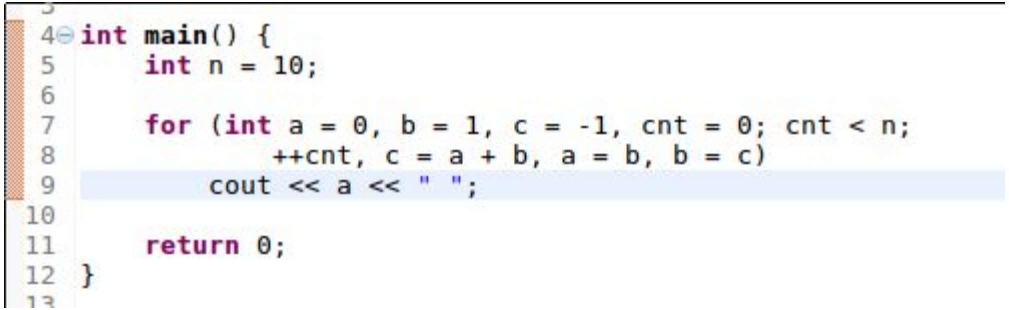


```
#include<iostream>
using namespace std;

int main() {
    int n = 10;

    int a = 0, b = 1;

    cout<<a<<" "<<b<<" ";

    for (int cnt = 2; cnt < n; ++cnt) {
        int c = a+b;
        a = b;
        b = c;

        cout<<c<<" ";
    }

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/
0 1 1 2 3 5 8 13 21 34
```

Fibonacci Sequence

- Same code but written in different way
  - 1) For loop can spread several lines
  - 2) Initialization can initialize several variables
  - 3) Step can changes several variables



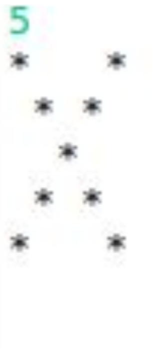
```
int main() {
    int n = 10;

    for (int a = 0, b = 1, c = -1, cnt = 0; cnt < n;
            ++cnt, c = a + b, a = b, b = c)
        cout << a << " ";

    return 0;
}
```

Homework

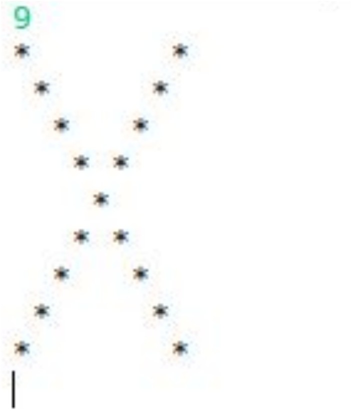
- Repeat all the last lecture homework, but using for loops

Homework 1: Printing X

- Read an Integer N, then print an X using * as following
  - N always odd



```
5
*   *
 * *
  *
 * *
*   *
```



```
9
*       *
 *     *
  *   *
   * *
    *
   * *
  *   *
 *     *
*       *
```

Homework 2: Find Special Pairs

- Count How many X, Y numbers such that
  - X in range [50-300]
  - Y in range [70-400]
  - X < Y
  - (X+Y) divisible by 7
- Output
  - 8040

Homework 3: Find all quadruples

- Count how many (a, b, c, d) with following property:
  - 1 <= a, b, c, d <= 200
  - a + b = c + d
- Code it once using 4 loops
  - How much time does it take!
- Code it once using 3 loops only
- Future: With hash tables, you can do it using 2 loops only

Homework 4: Is Prime?

- Read an integer N (< 500) and print YES if it is prime, otherwise NO
    - A prime number is greater than 1 AND cannot be formed by multiplying two smaller numbers.
        - In other words, number%whatever != 0
        - The first few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29.
- Input ⇒ Output
    - 13 ⇒ YES (only 1 * 13)
    - 12 ⇒ NO (E.g. 12 = 2 *6, so 12 can be divided by 2 or 6)

Homework 5: Print Primes

- Read integer N (<500), then print all prime numbers <= N
  - Output should be comma separated, as below
    - Don’t print comma after the last number
- Input ⇒ Output
  - 18 ⇒ 2,3,5,7,11,13,17
    - No comma after last number!

Homework 6: Digits sum in range

- Read three numbers N, A, B. Print the summation of the numbers between 1 and N whose sum of digits is between A and B.
- Input / Output
  - 20 2 5 ⇒ 84
    - Numbers whose sums of digits are between 2 and 5, are: 2,3,4,5,11,12,**13**,14, 20.
      - E.g. digits sum of 13 is 4 : which is between (2, 5)
  - 10 1 2 ⇒ 13
  - 100 4 16 ⇒ 4554

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً