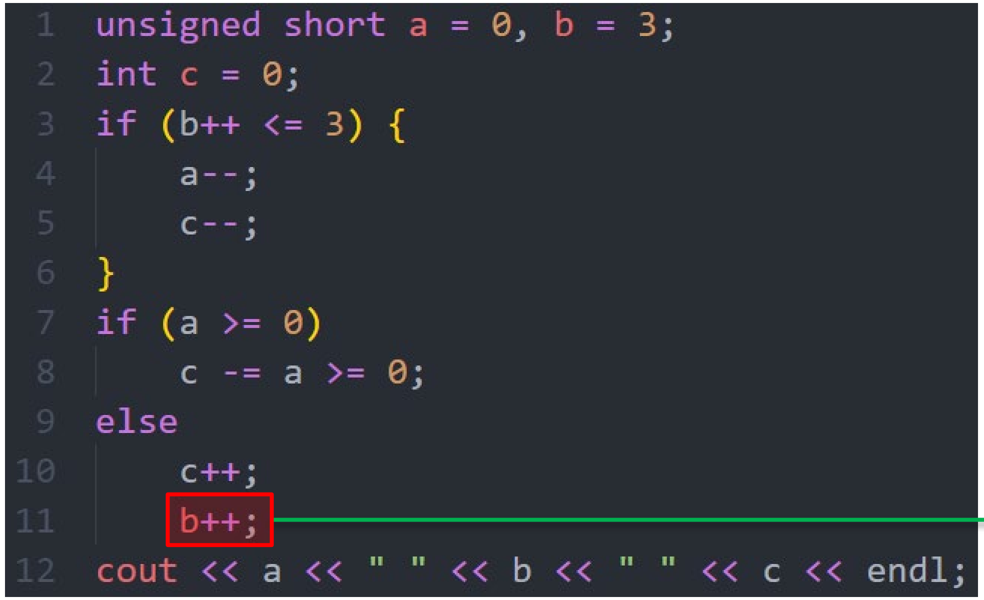
CSCI 1540

Fundamental Computing with C++

Tutorial 9
Quiz Solutions

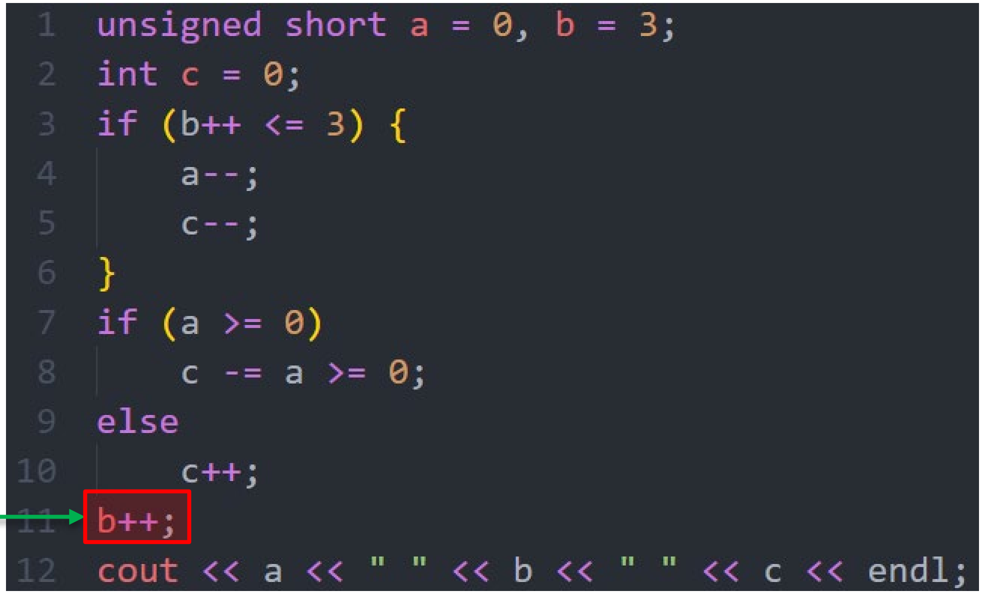
ZHANG Yong
SHB 902
zhangy@cse.cuhk.edu.hk

Problem 1(a)



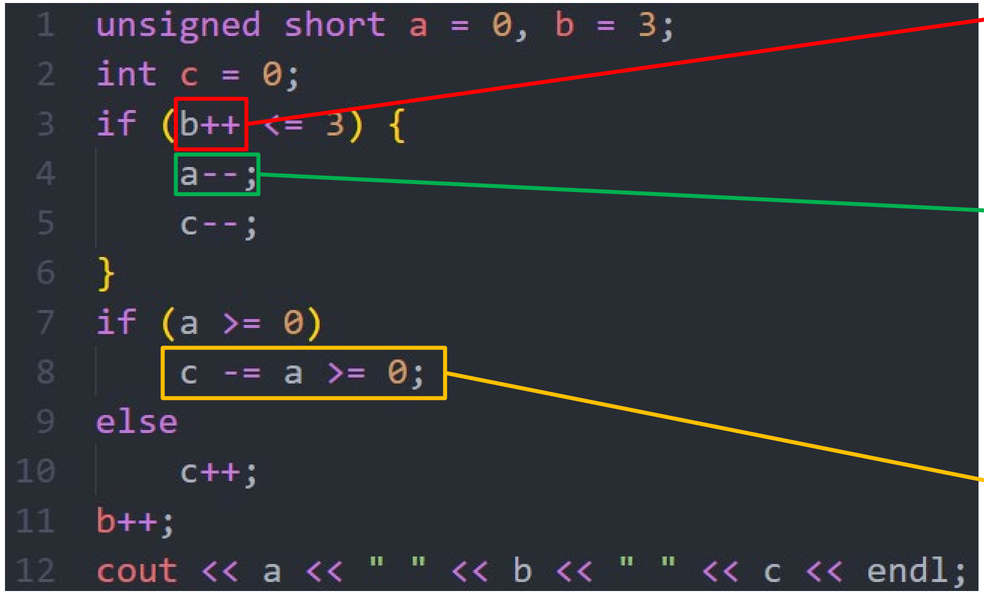
```
unsigned short a = 0, b = 3;
int c = 0;
if (b++ <= 3) {
    a--;
    c--;
}
if (a >= 0)
    c -= a >= 0;
else
    c++;
    b++;
cout << a << " " << b << " " << c << endl;
```

Proper indentation



```
unsigned short a = 0, b = 3;
int c = 0;
if (b++ <= 3) {
    a--;
    c--;
}
if (a >= 0)
    c -= a >= 0;
else
    c++;
b++;
cout << a << " " << b << " " << c << endl;
```

Problem 1(a)



```
unsigned short a = 0, b = 3;
int c = 0;
if (b++ <= 3) {
    a--;
    c--;
}
if (a >= 0)
    c -= a >= 0;
else
    c++;
b++;
cout << a << " " << b << " " << c << endl;
```

Return 3.
Afterwards, b becomes 4.

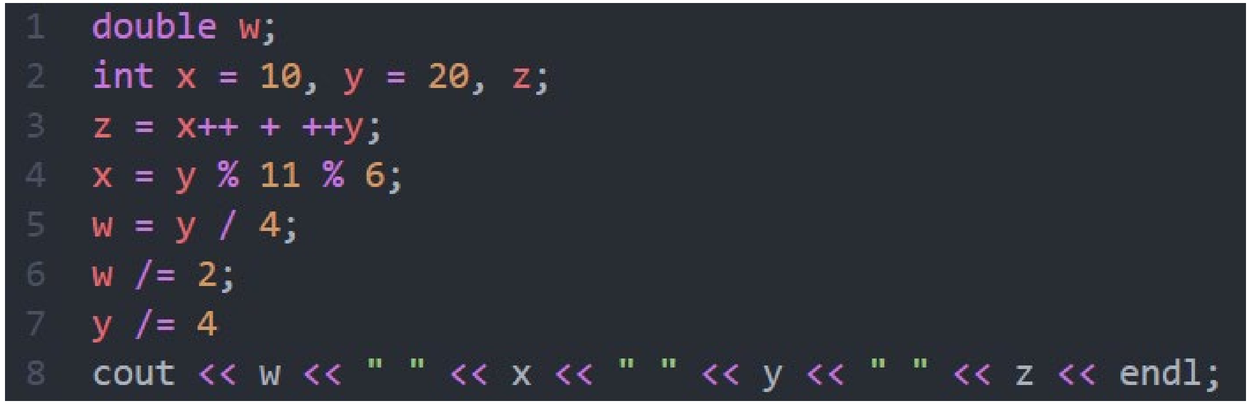
unsigned short is 16-bit and 2^{16} = 65536
a is *unsigned short* type
a is 0...65535
a = 0; a--; // a becomes 65535

Precedence: >= (higher), -= (lower)
c -= (a >= 0);

- int tmp = true; // tmp = 1
- int tmp = false; // tmp = 0

Output: 65535 5 -2

Problem 1(b)



```
1  double w;
2  int x = 10, y = 20, z;
3  z = x++ + ++y;
4  x = y % 11 % 6;
5  w = y / 4;
6  w /= 2;
7  y /= 4
8  cout << w << " " << x << " " << y << " " << z << endl;
```

Line 3

- x++: return 10, x becomes 11
- ++y: return 21, y becomes 21
- `z` becomes `31`

Line 4

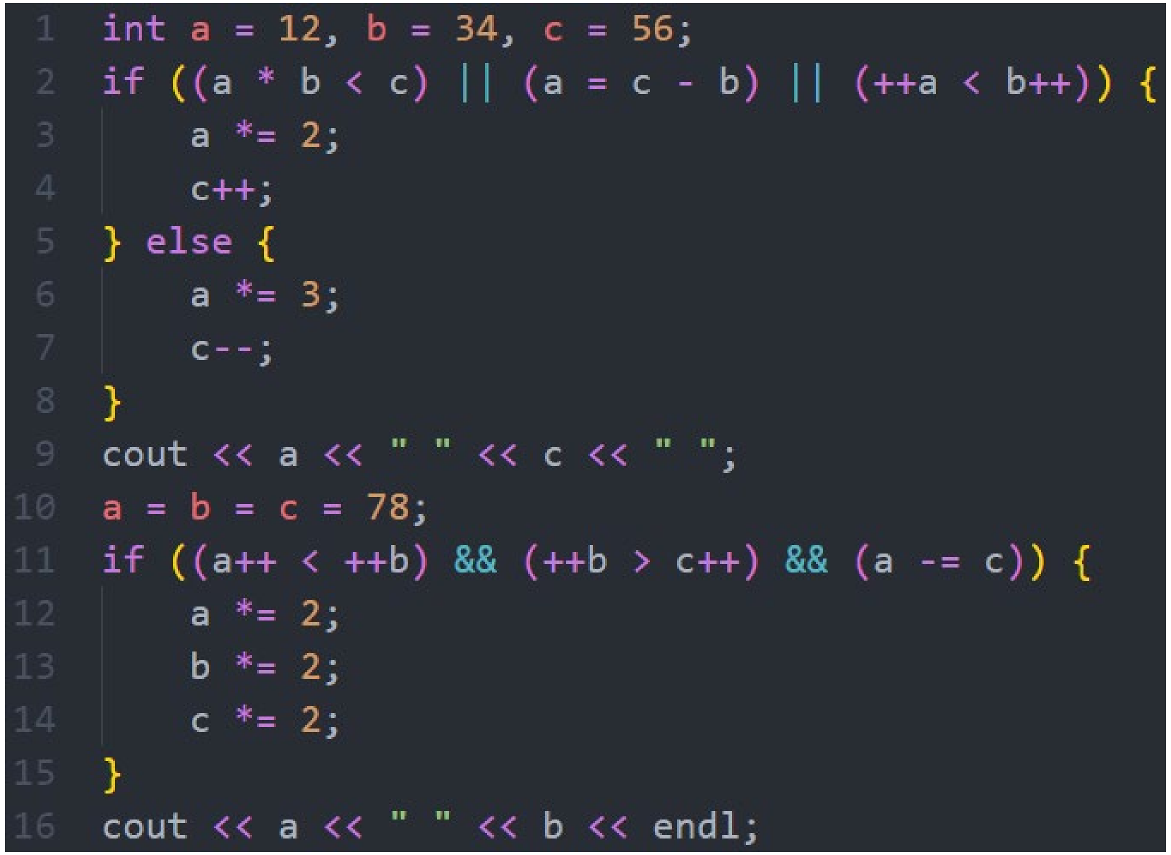
- % associativity: left-to-right
- `x = (y%11)%6;`
- `x` becomes `4`

Line 5, 6, 7

- y is int type: `y/4` equals `5;` `w` gets `5.0`
- w is double type: `5.0/2` equals `2.5`

Output: `2.5 4 5 31`

Problem 1(c)



```
 1  int a = 12, b = 34, c = 56;
 2  if ((a * b < c) || (a = c - b) || (++a < b++)) {
 3      a *= 2;
 4      c++;
 5  } else {
 6      a *= 3;
 7      c--;
 8  }
 9  cout << a << " " << c << " ";
10  a = b = c = 78;
11  if ((a++ < ++b) && (++b > c++) && (a -= c)) {
12      a *= 2;
13      b *= 2;
14      c *= 2;
15  }
16  cout << a << " " << b << endl;
```

Short-circuit evaluations

Line 2

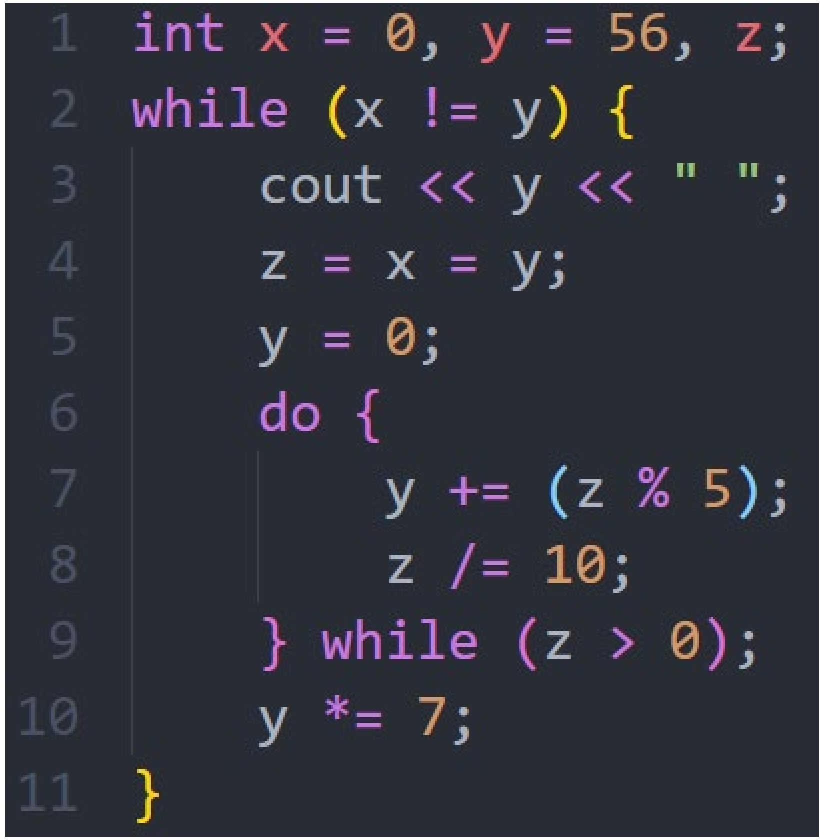
- a * b < c: 12 * 34 < 56 (false)
- a = c – b: 56 – 34 (true)
  - a becomes 22
- ++a < b++: NOT be executed

Line 11

- a++ < ++b: 78 < 79 (true)
  - a and b both become 79
- ++b > c++: 80 > 78 (true)
  - b becomes 80
  - c becomes 79
- a -= c: 79 - 79 (false)
  - a becomes 0

Output: 44 57 0 80

Problem 1(d)

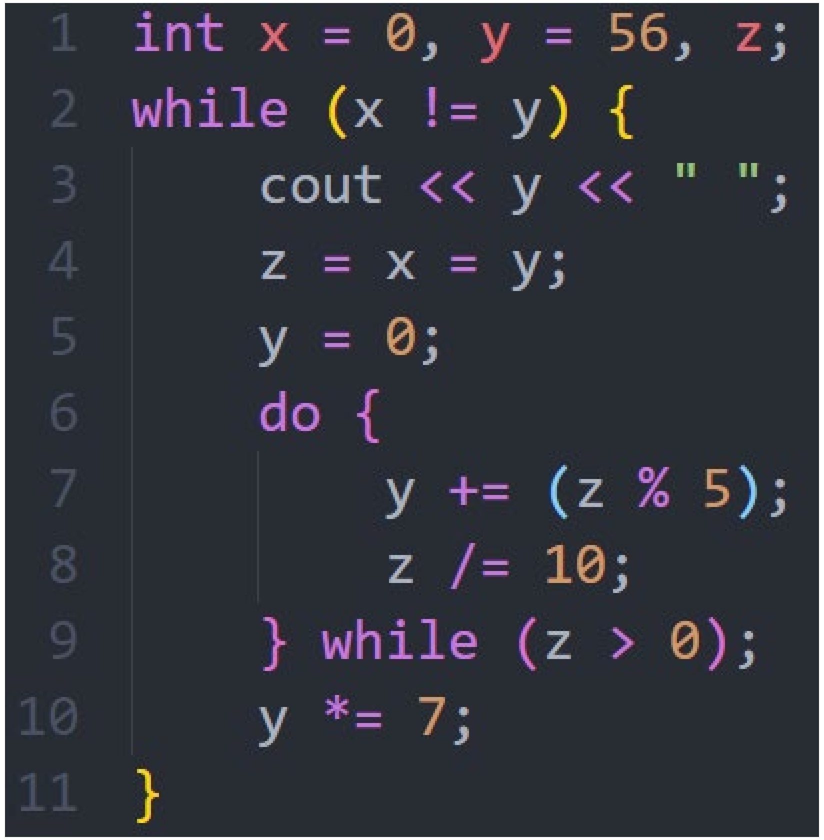


```
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 0, y = 56
true
print y = 56
z = x = y = 56
y = 0
y = 0 + 56 % 5 ➔ y = 1
z = 56 / 10 ➔ z = 5
true
y = 1 + 5 % 5 ➔ y = 1
z = 5 / 10 ➔ z = 0
false
y = 1 * 7 ➔ y = 7
```

Output: 56

Problem 1(d)

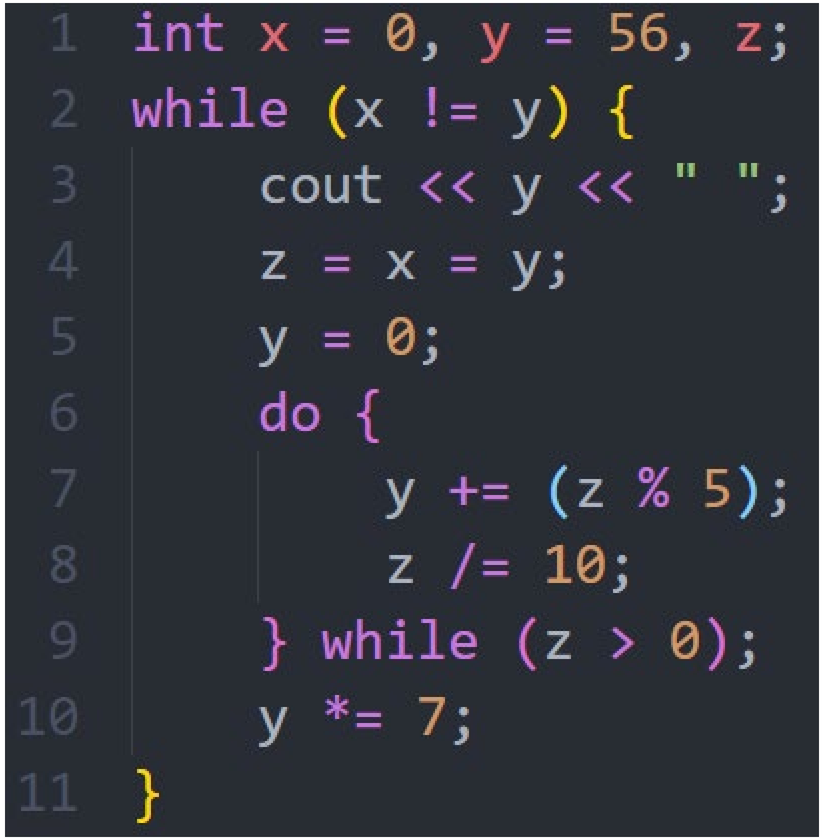


```
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 56, y = 7, z = 0
true
print y = 7
z = x = y = 7
y = 0
y = 0 + 7 % 5 ➔ y = 2
z = 7 / 10 ➔ z = 0
false
y = 2 * 7 ➔ y = 14
```

Output: 56 7

Problem 1(d)

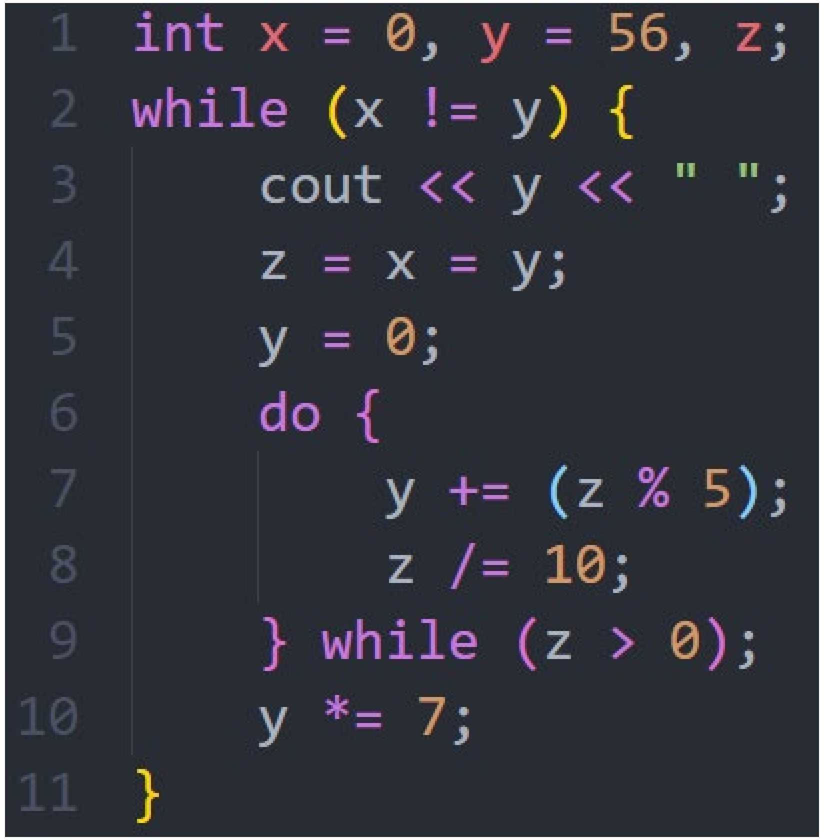


```
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 7, y = 14, z = 0
true
print y = 14
z = x = y = 14
y = 0
y = 0 + 14 % 5 ➔ y = 4
z = 14 / 10 ➔ z = 1
true
y = 4 + 1 % 5 ➔ y = 5
z = 1 / 10 ➔ z = 0
false
y = 5 * 7 ➔ y = 35
```

Output: 56 7 14

Problem 1(d)

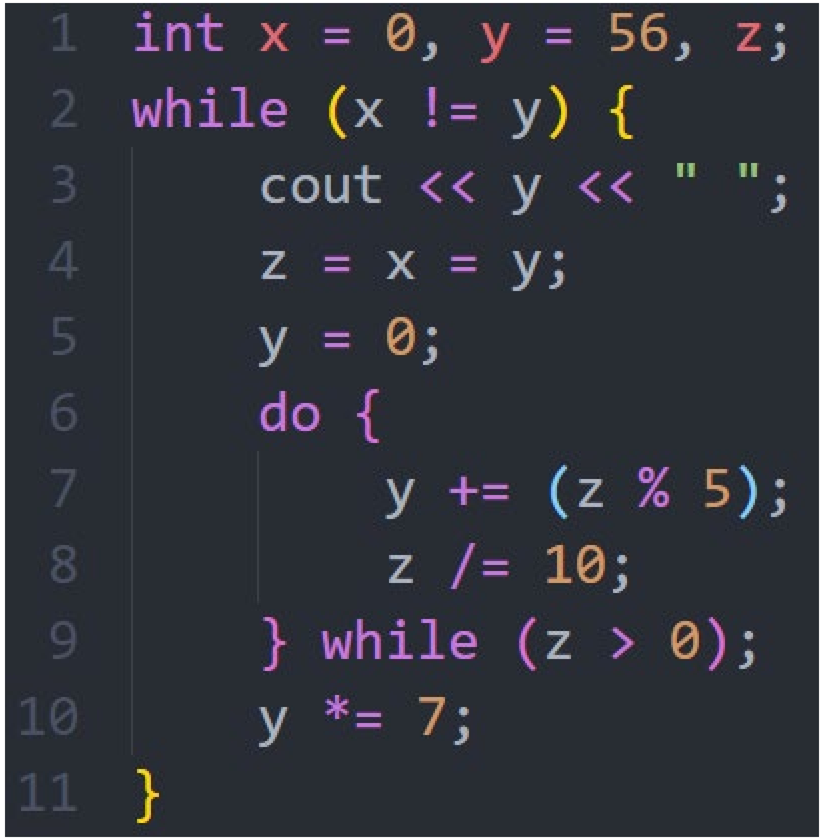


```cpp
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 14, y = 35, z = 0
true
print y = 35
z = x = y = 35
y = 0
y = 0 + 35 % 5 ➔ y = 0
z = 35 / 10 ➔ z = 3
true
y = 0 + 3 % 5 ➔ y = 3
z = 3 / 10 ➔ z = 0
false
y = 3 * 7 ➔ y = 21
```

Output: 56 7 14 35

Problem 1(d)

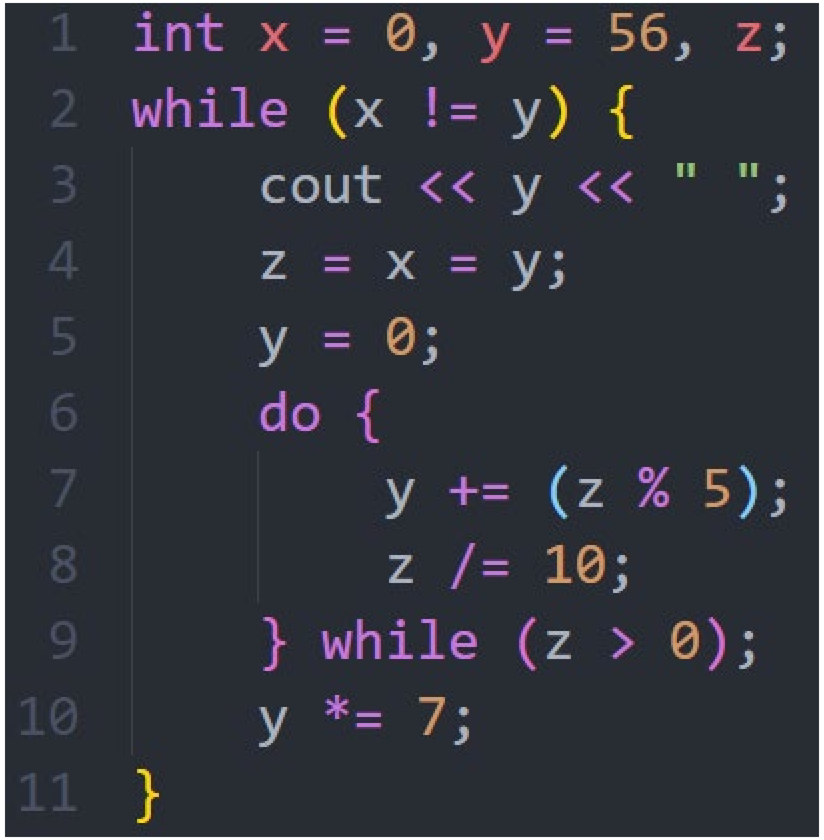


```
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 35, y = 21, z = 0
true
print y = 21
z = x = y = 21
y = 0
y = 0 + 21 % 5 ➔ y = 1
z = 21 / 10 ➔ z = 2
true
y = 1 + 2 % 5 ➔ y = 3
z = 2 / 10 ➔ z = 0
false
y = 3 * 7 ➔ y = 21
```

Output: 56 7 14 35 21

Problem 1(d)



```
int x = 0, y = 56, z;
while (x != y) {
    cout << y << " ";
    z = x = y;
    y = 0;
    do {
        y += (z % 5);
        z /= 10;
    } while (z > 0);
    y *= 7;
}
```

```
x = 21, y = 21, z = 0
false
```

END

Output: 56 7 14 35 21

Problem 2

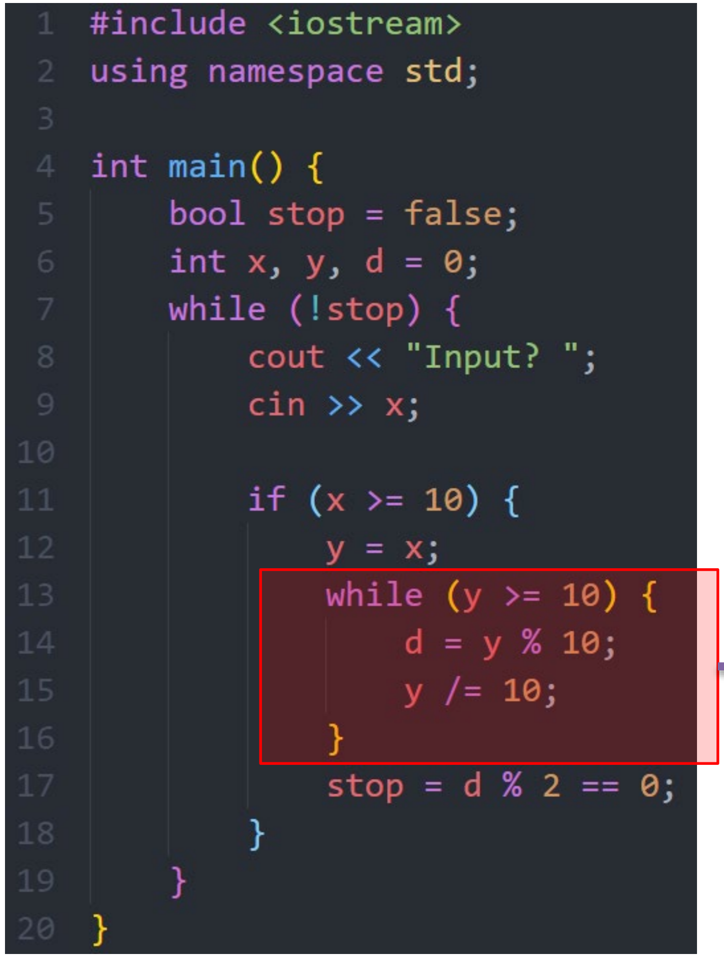
Write a code fragment that repeatedly obtains integer user inputs, until the input is positive, has at least two digits, *and* its second leftmost digit is even. You can assume that *the users always enter integers within the int range*. The following shows a sample run. The numbers after '?' are user inputs.

```
Input? -465
Input? 276
Input? -465
Input? 123
```

Problem 2

- Repeatedly obtain integers
  - A loop (outer)
- The positive value + at least two digits
  - If statement: `value >= 10`
- Extract the second leftmost digit
  - A loop (inner)
  - %, /

Problem 2

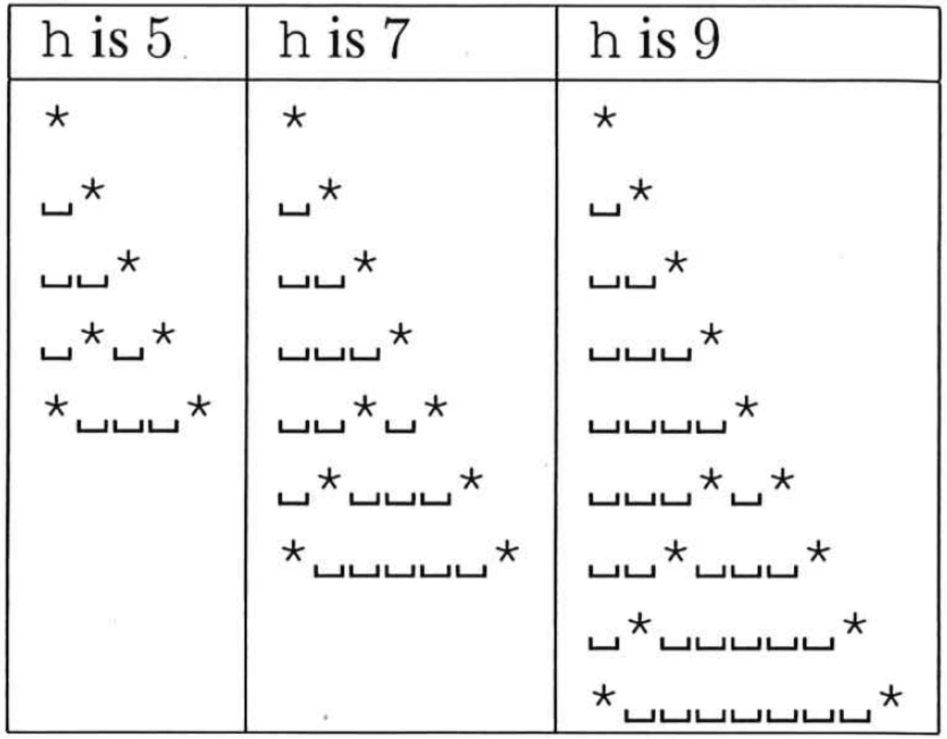


```
#include <iostream>
using namespace std;

int main() {
    bool stop = false;
    int x, y, d = 0;
    while (!stop) {
        cout << "Input? ";
        cin >> x;

        if (x >= 10) {
            y = x;
            while (y >= 10) {
                d = y % 10;
                y /= 10;
            }
            stop = d % 2 == 0;
        }
    }
}
```

When getting out of the loop, *d* is the second leftmost digit.

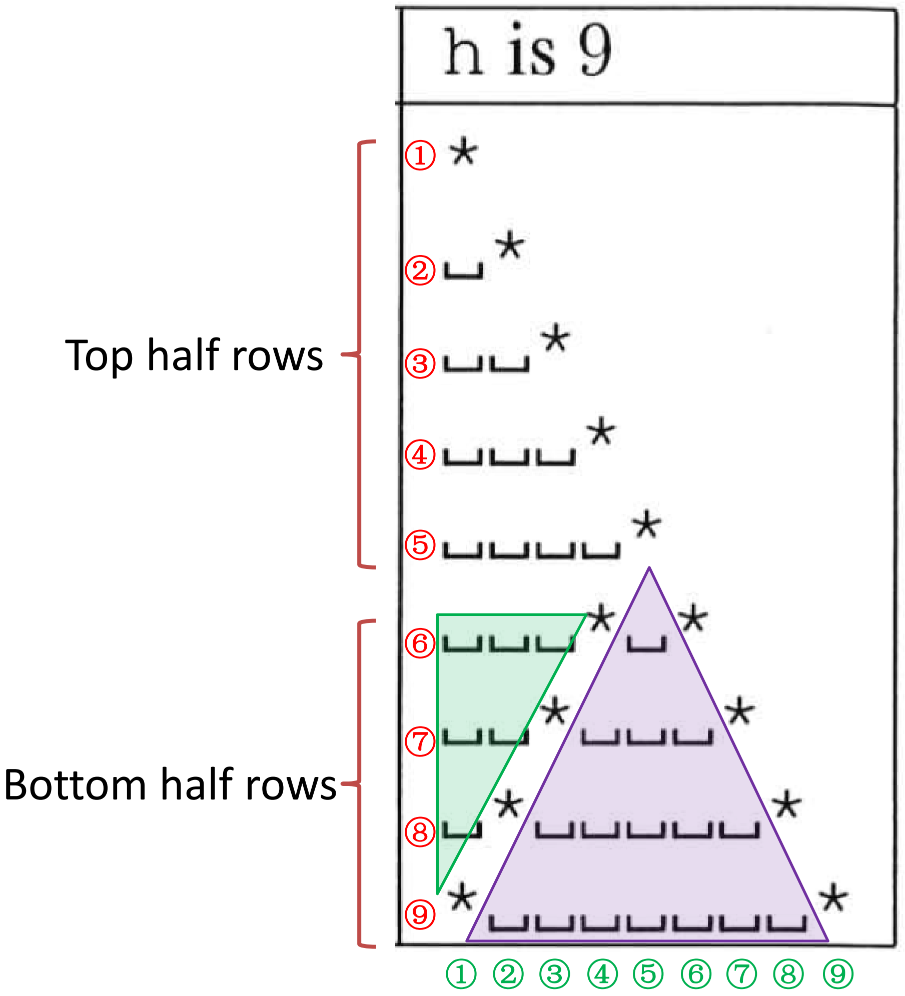
Problem 3

Write a code fragment that prints a λ symbol of height h. You can assume that h is odd and $h \geq 5$.



h is 5	h is 7	h is 9
*	*	*
␣*	␣*	␣*
␣␣*	␣␣*	␣␣*
␣*␣*	␣␣␣*	␣␣␣*
␣␣␣	␣␣*␣*	␣␣␣␣*
	␣*␣␣␣*	␣␣␣*␣*
	␣␣␣␣␣	␣␣*␣␣␣*
		␣*␣␣␣␣␣*
		␣␣␣␣␣␣␣

Problem 3



Nested loop

- Outer loop: iterate rows
- Inner loop: iterate columns

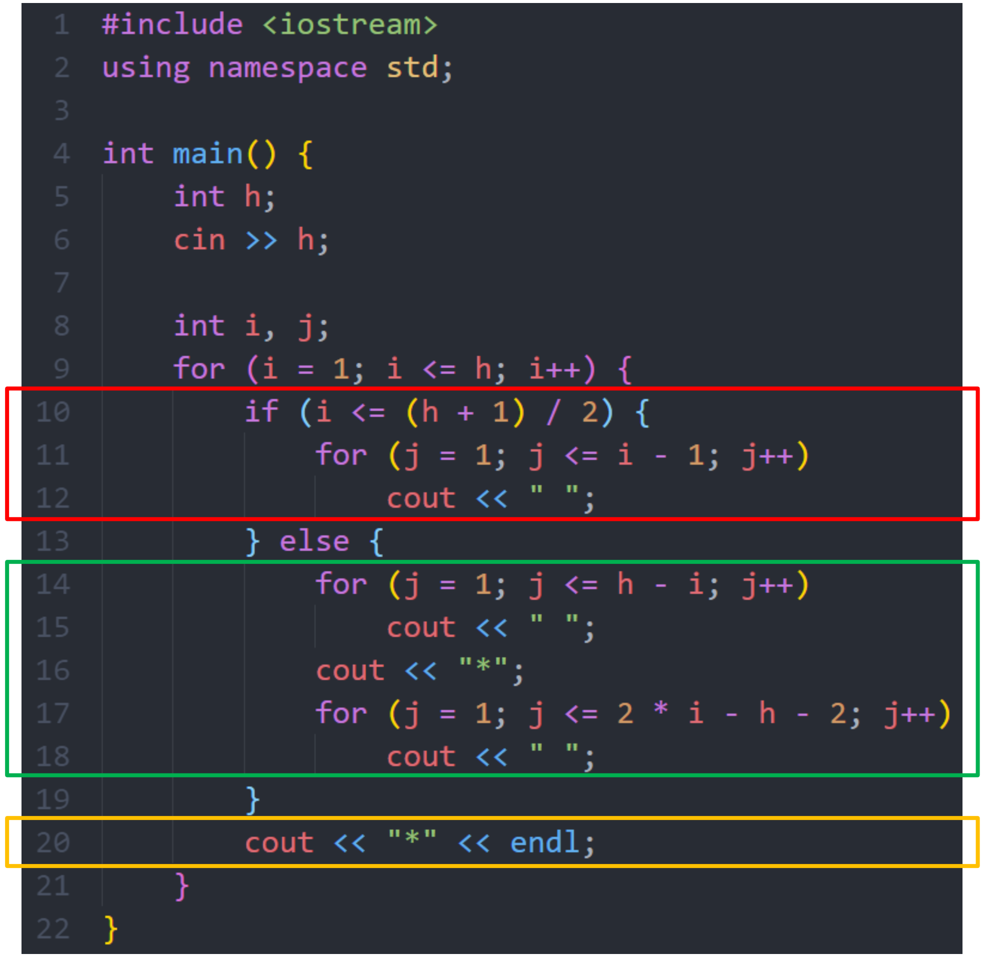
Top: only one ⋆ for each row

- ⋆ pos: *row_label == col_label*
- ␣ numbers: *row_lable - 1*

Bottom: two ⋆ for each row

- 2nd ⋆ pos: *row_label == col_label*
- 1st ⋆ pos: *h – row_lab + 1*
- ␣ numbers of 1st part: *h – row_lab*
- ␣ numbers of 2nd part: *2 * row_label – h – 2*

Problem 3



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

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

    int i, j;
    for (i = 1; i <= h; i++) {
        if (i <= (h + 1) / 2) {
            for (j = 1; j <= i - 1; j++)
                cout << " ";
        } else {
            for (j = 1; j <= h - i; j++)
                cout << " ";
            cout << "*";
            for (j = 1; j <= 2 * i - h - 2; j++)
                cout << " ";
        }
        cout << "*" << endl;
    }
}
```

Top half rows

Bottom half rows

The last * in each row

Q & A