CSCI 1540
Fundamental Computing with C++

Quiz Review

ZHANG Yong

SHB 902

zhangy@cse.cuhk.edu.hk

Reminder of Quiz

- Date: Mon 21 Oct 2019
- Time: 12:30pm - 13:30pm (1 hour)
- Venue: LSB LT1
- Scope: lec01 – lec06

Quiz Review

Questions & Answers

Problem 1 (a)

```
int x = 0;
x = 4++;
cout << x;
```

Is this code valid? What is the output of this code?

Analysis: 4 is a constant. We can only apply "++" and "--" operators to variables.

Result: Invalid

Problem 1 (b)

What is the output of this code?

```
int x = 10, y = 5, z = 0;
for (int i = 0; i < 3; i++)
{
        z = x++ - --y;
}
cout << x <<' ' << y << ' '<< z;
```

Prefix: “++x” “--x”	Postfix: “x++” “x--”
Increase/decrease the operand	
Return the value after change	Return the value before change

Result: 13 2 10

Loop 1:
x: return 10 and increase to 11
y: decrease to 4 and return 4
z: 10-4=6
Loop 2:
x: return 11 and increase to 12
y: decrease to 3 and return 3
z: 11-3=8
Loop 3:
x: return 12 and increase to 13
y: decrease to 2 and return 2
z: 12-2=10

Problem 1 (c)

```
int a = 4;
if (a > 0)
    { cout << “A”;
      cout << “B”; }
else
      cout << “C”;
      cout << ‘D’;
```

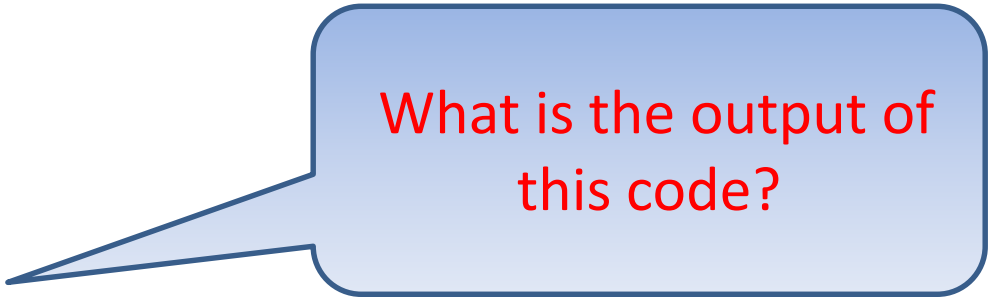
Is this code valid?
What is the output of this code?

Analysis: The syntax of if-else statement only allows one single statement after the condition and before the "else" keyword.
However, here we have two statements. How to make it valid?

Result: Invalid

Problem 1 (c)

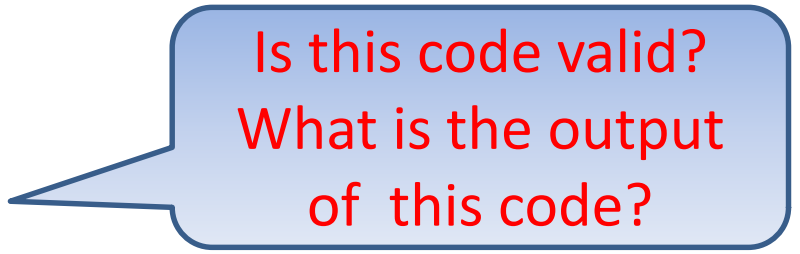
```
int a = 4;
if (a > 0)
        cout << “A”;
else
        cout << “C”;
        cout << ‘D’;
```



Result: AD

Problem 1 (d)

```
cout << "//////   \
          \\\\\\";
```



Analysis: String constant must be enclosed in double quotes within one line.

Result: Invalid

How to make it valid? String constant can extend to more than a single line of code by putting a backslash sign (\) at the end of each unfinished line.

Problem 2

```
int A, B;
cin >> A >> B;
```

```
/*
     Output all the odd numbers between A and B in
     descending order.
*/
```

Problem 2

- Compare A and B to see which one is larger
- Loop from the larger number to the smaller one
  - Start from the largest number and decrease 1 each time and only output the odd numbers
    - Odd number % 2 == 1

    Or
  - Start from the largest **ODD** number and decrease 2 each time and output all numbers

Problem 2

```
int A, B;   /* declare two variables to define the range */
int temp; /* declare a variable for swap operation */
int i;  /* declare a varialbe for loop index */
cin >> A >> B;  /* read two variables to define the output range */
```

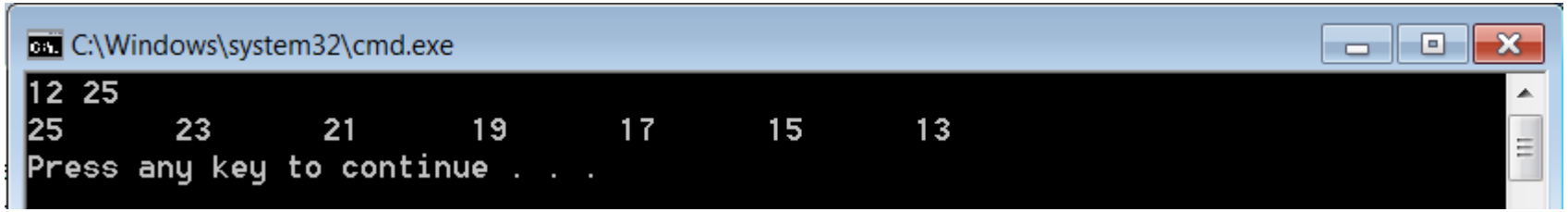
```
if (A>B)  /* if A>B, swap A and B */
{
    temp=A;
    A=B;
    B=temp;
}
```

```
/* since B > A, start from A, decrease 1 each time, output the odd numbers */
for (i = B; i >= A; i--)
    if (i % 2 == 1)
        cout << i;
cout << endl;   /* new line */
```

Other options, e.g., if(i%2)

Problem 2

```cpp
int main() {
    int A, B;         /* declare two variables to define the range */
    int temp;         /* declare a variable for swap operation */
    int i;            /* declare a variable for loop index */
    cin >> A >> B;    /* read two variables to define the output range */

    if (A > B) {      /* if A > B, swap A and B */
        temp = A;
        A = B;
        B = temp;
    }

    /* since B > A, start from B, decrease 1 each time, output the odd numbers */
    for (i = B; i >= A; i--)
        if (i % 2 == 1)
            cout << i << "\t";
    cout << endl;     /* new line */

    return 0;
}
```



Problem 2

- Anything wrong for the previous codes?
  - What if A or B is negative?
    - A = -2, B = -9; ->NO OUTPUT
    - A = -2, B = 9; ->missing some Odd numbers

- Odd number <=> (x % 2 != 0)
- Even number <=> (x % 2 == 0)

Problem 3

int A, B;

cin >> A >> B;

```
/*
Output the total number of integers in [A,B] that are
fully divisible by 3 or 7 or 11.
*/
```

Problem 3

- Define a variable as the counter and set it to 0
- Use a loop to scan all the numbers in range
  - If the number is divisible by 3, 7 or 11, add 1 to the counter
    - (number % 3 == 0) || (number % 7 == 0) || (number % 11 == 0)
    - counter++

Problem 3

```
int A, B;   /* declare two variables to define the range */
int counter=0; /* declare a variable to record all the numbers matching the requirement */
int i;  /* declare a varialbe for loop index */
```

```
cin >> A >> B;  /* read two variables to define the output range */
```

```
/* scan the number in range [A, B], and if the number is divisible by 3, 7 or 11, counter++ */
for (i = A; i <= B; i++)
    if (i % 3 == 0 || i % 7 == 0 || i % 11 == 0)
        counter++;
cout << counter <<  endl; /* print the number */
```

Why? Think about 21

Remark:

```
if (i%3==0)
    counter++;
if (i%7==0)
    counter++;
if (i%11==0)
    counter++;
```

Does not work!

Problem 3

```cpp
int main() {
    int A, B;       /* declare two variables to define the range */
    int counter = 0;/* declare a variable to record the numbers matching the requirement */
    int i;          /* declare a variable for loop index */
    cin >> A >> B;  /* read two variables to define the output range */


    /* scan the number in range [A, B], and if the number is divisible by 3, 7 or 11, counter++ */
    for (i = A; i <= B; i++)
        if (i % 3 == 0 || i % 7 == 0 || i % 11 == 0)
            counter++;
    cout << counter <<  endl; /* print the number */

    return 0;
}
```

Problem 4

```
/*
     Read a sequence of integers from the user until
a zero is encountered in the input sequence, and
output "YES" only if there exists two consecutive
integers in the input sequence in which their sum is
equal to 100.

     For example, if the input sequence is
          1 2 3 97 99 0.
Try the sum of two consecutive integers:
1+2=3, 2+3=5, 3+97=100.
Your code should output YES because 3 + 97 = 100.
*/
```

Problem 4

- Use a loop to read only one number each time
  - When the current number is 0, jump out of the loop
  - Check if the sum of the current number and the last number is 100
    - Need to store the last number as well
    - In the first round, there's no last number
    - Use a boolean variable to record "YES"

Problem 4

```
int flag;       /* boolean variable, to record the state whether there exists
                   two consecutive integers whose sum is equal to 100 */
```

```
int cur;        /* declare a variable to record the current input number */
int prev;       /* declare a variable to record the previous integer */
```

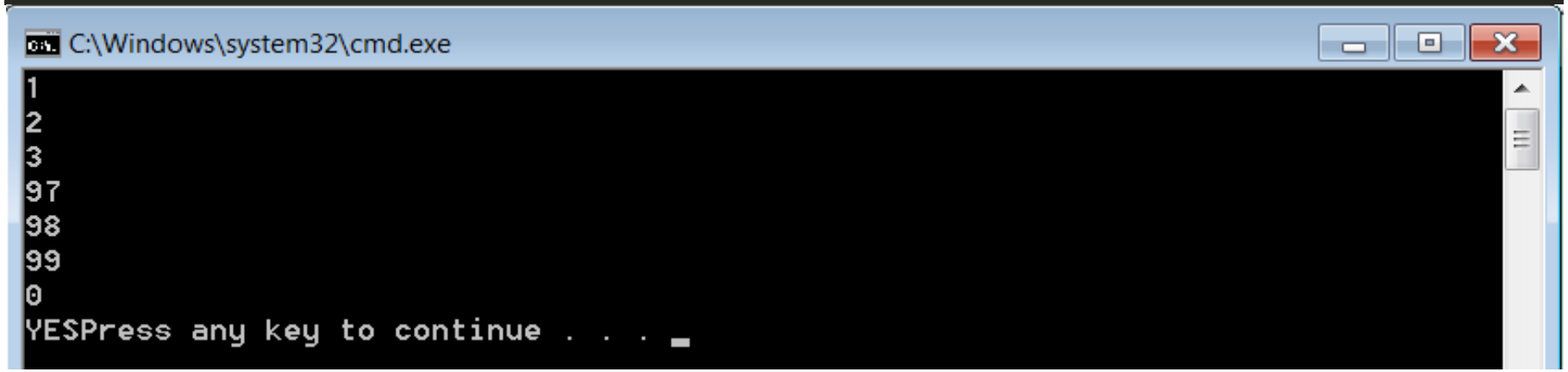
```
cin >> cur;     /* read the first input */
```

```
while (cur != 0) { /* check the curent number, if it is zeor, jump out the loop */
    prev = cur; /* store the current number as previous one */
    cin >> cur; /* read a new number */
    if (prev + cur == 1
        flag = 1;
}
```

```
if (flag)       /* check the flag, if t          ut "YES" */
    cout << "YES";
```

Other options, e.g., while(cur)

```cpp
int main() {
    int flag;       /* boolean variable, to record the state whether there exists
                       two consecutive integers whose sum is equal to 100 */
    int cur;        /* declare a variable to record the current input number */
    int prev;       /* declare a variable to record the previous integer */
    cin >> cur;     /* read the first input */

    while (cur != 0) { /* check the curent number, if it is zeor, jump out the loop */
        prev = cur; /* store the current number as previous one */
        cin >> cur; /* read a new number */
        if (prev + cur == 100)
            flag = 1;
    }

    if (flag)       /* check the flag, if true, output "YES" */
        cout << "YES";

    return 0;
}
```



Problem 5

- Write a segment of code that reads an integer *N* and *M* from the user and prints a rectangular checker board consisting of *N* rows and *M* columns of 2x2 squares using alternating pattern made up of characters '#' and '.'.
- Note: In this problem, every row and column is 2 character wide. And, you may assume *1≤N, M≤20.*

Problem 5--Analysis

- N=1, M=1

```
##
##
```

N=3, M=1

```
##
##
..
..
##
##
```

N=3, M=3

```
##..##
##..##
..##..
..##..
##..##
##..##
```

N=4, M=5

```
##..##..##
##..##..##
..##..##..
..##..##..
##..##..##
##..##..##
..##..##..
..##..##..
```

Problem 5--Analysis

N=4, M=5

```
1##..##..##
2##..##..##
3..##..##..
4..##..##..
5##..##..##
6##..##..##
7..##..##..
8..##..##..
```

1) row 1 is equal to row 2, row 3 is equal to row 4→ for each iteration of N, display each row twice!!

2) on each row, the 1st character is equal to the 2nd character, the 3rd character is equal to 4th character →for each iteration of M, display the same character twice!!

3) the two patterns "##" and ".." should alternate each other. Observed that on the i-th iteration of N, and the j-th iteration of M, when (i+j) is even, we display ##, otherwise we display ..→(i+j)/2==0 <--> "##"

Problem 5--Solution

```
int N, M;
cin >> N >> M;
for(int i=0; i<N; i++) {
    for(int j=0; j<M; j++){
        if((i+j)%2==0)
            cout<<"##";
        else
            cout<<"..";
    }
cout << endl;
```

print out the odd rows

```
for (int j=0; j<M; j++) {
    if((i+j)%2==0)
        cout<<"##";
    else
        cout<<"..";
    }
cout << endl;
}
```

print out the even rows

Problem 6 (a)

- Show the output of the following code:

```
for (int i = 0; i <= 10; i++) {
   if (i % 2) {
      if (i % 5 ==0)
            break;
      cout << i << " ";
   }
}
```

→true only when i is odd

→true only when i is divisible by 5

What is the usage of break?

It will terminate the for loop!!

Output: 1 3

Problem 6 (b)

- Show the output of the following code when the input sequence is: 3 6 5 2 8 9 7 10 -1 12

```
int x=0, y=0;
for(int i=0; i<6; i++) {
        cin>>x;
        if (x<y)
          continue;
        cin>>y;
        if(x<y)
          continue;
        cout<<x<< “  ”<<y<<endl;
}
```

Analysis: continue
the execution will jump to i++ continuing the next iteration, statements below continue will not be executed.

Result: 10 -1

Start
x = 0
y = 0
i = 0
x = 3
y = 6
i = 1
x = 5
i = 2
x = 2
i = 3
x = 8
y = 9
i = 4
x = 7
i = 5
x = 10
y = -1
"10 -1"
i = 6
End

Problem 7

- Write a code fragment that reads an integer N from the user (assume N is positive) and prints the first N integers in the following number sequence:

  1 3 -5 -7 9 11 -13 -15 17 19 -21 -23 ...

  (odd numbers; changing signs every two values)

Problem 7

Odd number formula: x= 2*i+1

- Number: 1 3 -5 -7 9 11 -13 -15 17 19 -21 -23
- Index: 0 1 2 3 4 5 6 7 8 9 10 11
- Flag: + + - - + + - - + + - -

1. Use a variable i to store index, and output odd number = 2*i+1, where i is in range [0, N-1]

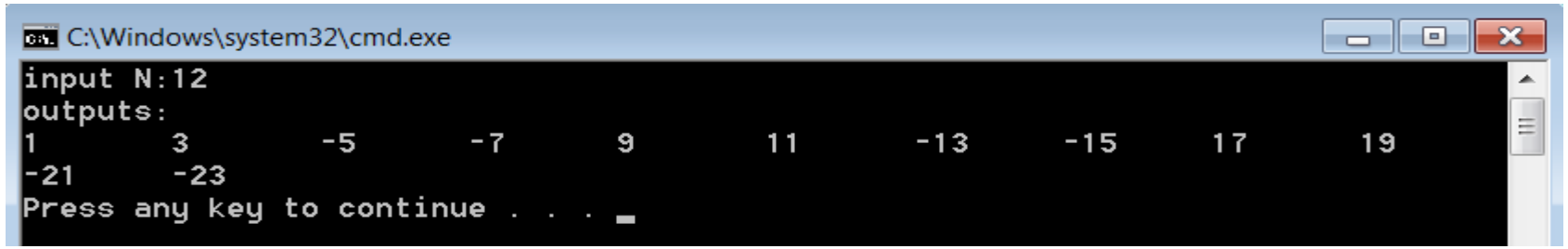
2. Use a variable flag to record the sign; when i is an even number, change the flag ; initial value is minus 1

Problem 7

```
int N;          /* store the number of inputs */
int odd;        /* store odd number */
int flag = -1;  /* sentinel variable */
int i;          /* variable for loop index */

cout << "input N:";
cin >> N;
cout << "outputs:" << endl;

for (i = 0; i < N; i++) {
    odd = 2 * i + 1;    /* get odd number */

    if (i % 2 == 0)     /* when i is even number, reverse flag */
        flag = -flag;

    if (flag == 1)      /* output number according to flag value */
        cout << odd << "\t";
    else
        cout << -odd << "\t";
}
```

Pay attention to the range

```c
int main() {
    int N;          /* store the number of inputs */
    int odd;        /* store odd number */
    int flag = -1;  /* sentinel variable */
    int i;          /* variable for loop index */

    cout << "input N:";
    cin >> N;
    cout << "outputs:" << endl;

    for (i = 0; i < N; i++) {
        odd = 2 * i + 1;    /* get odd number */

        if (i % 2 == 0)     /* when i is even number, reverse flag */
            flag = -flag;

        if (flag == 1)      /* output number according to flag value */
            cout << odd << "\t";
        else
            cout << -odd << "\t";
    }

    return 0;
}
```



Problem 8

- Write a code fragment that reads an integer N from the user (assume N is positive) and prints a parallelogram consisting of N rows with alternating patterns and the number of symbols in each row should be N. (Print 1st row with '*', 2nd row with '$', 3rd row with '*', 4th row with '$', and so on.)

Problem 8

- Input samples:
  - when N = 6, your output should be:

```
******
 $$$$$$
  ******
   $$$$$$
    ******
     $$$$$$
```

  - when N = 7, your output should be:

```
*******
 $$$$$$$
  *******
   $$$$$$$
    *******
     $$$$$$$
      *******
```

Problem 8

- 1. Use a loop to output N rows.
- 2. When index is even, output ‘*’, otherwise ‘$’.
- 3. Use a loop to spare spaces in the number of index in each row.
- 4. Use a loop to print symbols in the number of N.

Problem 8

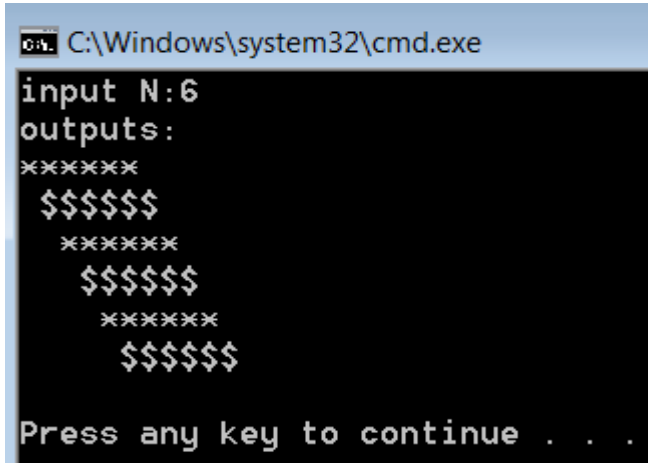
```
int N;          /* store the number of inputs */
int i;          /* a variable for loop index to output N rows*/
int j;          /* a variable for loop index to output spaces*/
```

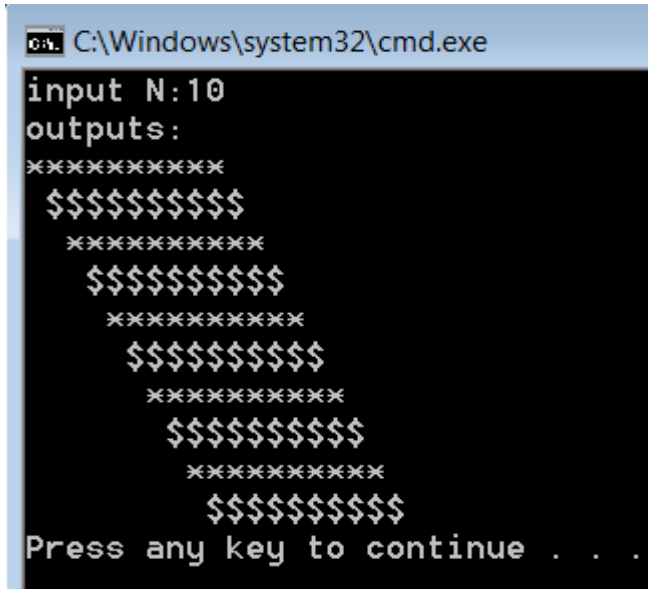
```
cout << "input N:";
cin >> N;
cout << "outputs:" << endl;
```

```
for (i = 0; i < N; i++) {       /* output N rows */
    if (i % 2 == 0) {           /* even row */
        for (j = 0; j < i; j++) /* output spaces */
            cout << " ";
        for (j = 0; j < N; j++) /* output star */
            cout << "*";
        cout << endl;           /* new row */
    } else {
        for (j = 0; j < i; j++) /* output spaces */
            cout << " ";
        for (j = 0; j < N; j++) /* output dollar */
            cout << "$";
        cout << endl;           /* new row */
    }
}
```

- Pay attention to the range of the loop

```cpp
int main() {
    int N;          /* store the number of inputs */
    int i;          /* a variable for loop index to output N rows*/
    int j;          /* a variable for loop index to output spaces*/

    cout << "input N:";
    cin >> N;
    cout << "outputs:" << endl;

    for (i = 0; i < N; i++) {       /* output N rows */

        if (i % 2 == 0) {           /* even row */
            for (j = 0; j < i; j++) /* output spaces */
                cout << " ";
            for (j = 0; j < N; j++) /* output star */
                cout << "*";
            cout << endl;           /* new row */
        } else {
            for (j = 0; j < i; j++) /* output spaces */
                cout << " ";
            for (j = 0; j < N; j++) /* output dollar */
                cout << "$";
            cout << endl;           /* new row */
        }
    }

    return 0;
}
```





Problem 9(a)

Show the output produced by the following segment of code:

```
for (i = 0; i < 10; i++){ \\ loop from 0 to 9
    if (i % 3 == 0) \\ i=0,3,6,9, go next iteration
        continue;
    cout << i << endl; \\ print i
}
```

Output:

```
1
2
4
5
7
8
```

Problem 9(b)

Show the output of the following segment of code:

```
int i, j;
for (i = 0; i <= 4; i++) {
  if (i % 2 == 0)
    continue;

  for (j = i; j <= 4; j++) {

    if (j % 2 == 0)
      break;

    cout << "A: " << i << " " << j << endl;
  }
  cout << "B: " << i << " " << j << endl;
}
cout << "C: " << i << " " << j << endl;
```

if i = 0,2,4, go to next iteration

When j first becomes an even number, the inner loop ends.

Problem 9(b)

Show the output of the following segment of code:

```
int i, j;
for (i = 0; i <= 4; i++) {
  if (i % 2 == 0)
    continue;

  for (j = i; j <= 4; j++) {

    if (j % 2 == 0)
      break;

    cout << "A: " << i << " " << j << endl;
  }
  cout << "B: " << i << " " << j << endl;
}
cout << "C: " << i << " " << j << endl;
```

Loop i = 0:

i % 2 == 0, thus we go to next iteration.

Loop i = 1:

j from 1 to 4. When j = 1, inner loop prints:

A: 1 1

j = 2, inner loop ends. Outer loop prints:

B: 1 2

Loop i = 2:

i % 2 == 0, go to next iteration.

Loop i = 3:

j from 3 to 4. When j = 3, inner loop prints:

A: 3 3

j = 4, inner loop ends. Outerloop prints:

B: 3 4

Loop i = 4:

i % 2 == 0, thus outer loop ends.

But now, i has become 5! Finally print

C: 5 4

Problem 10

- Consider this operation on a positive integer *x*: "*If x is even, divide it by two* $(\frac{x}{2})$. *If x is odd, triple it and add one* $(3x + 1)$." Start with any positive integer, and follow this operation repeatedly. The process should finally reach the number 1. E.g., starting with 12, we get the sequence 12, 6, 3, 10, 5, 16, 8, 4, 2, 1, which involves nine steps. The largest value in the sequence is 16. Write a code fragment that reads an integer to start this process and print out the sequence. The following shows two sample runs. The numbers after '?' are user input, which are assumed to be positive integers.

Input? 12 12 6 3 10 5 16 8 4 2 1 Steps = 9 Largest = 16	Input? 26 26 13 40 20 10 5 16 8 4 2 1 Steps = 10 Largest = 40

```
int n; // You can declare extra variables below
cout << "Input? ";
cin >> n; // Assume n > 0
……
```

Problem 10

- Key points
  - Check out odd/even (… % 2 == 0)
  - Form a sequence with the loop (i.e., update n (or relevant vars) for the next round)
  - Record steps
  - Keep track of the largest value

Problem 10

- Solution

```
int n; // You can declare extra variables below
cout << “Input? ”;
cin >> n; // Assume n > 0

int steps = 0, max;
max = n;
while (n > 1) {
        cout << n << “ ”;
        if (n % 2 == 0)
                n /= 2;
        else
                n = 3 * n + 1;
        steps++;
        if (n > max)
                max = n;
}
cout << “1” << endl;
cout << “Steps = ” << steps << endl;
cout << “Largest = ” << max << endl;
```

Good Luck to Your Quiz