Problem 1 Write a complete C++ program that asks the user to type a big integer n. It should then ask 4 times for the user to type a smaller value of n and then terminate the program. However, if the user ever enters a value that is not smaller it should immediately say *Goodbye* and terminate the program.

Partial credit will be given for programs that perform some of the required steps but excessively long or complicated programs will lose credit.

Examples of two sample runs of the program:

```
venus> ./a.out
Type a big integer n:   100
Type a smaller value of n: 20
Type a smaller value of n: 10
Type a smaller value of n: 2
Type a smaller value of n: 1
venus>
```

```
venus> ./a.out
Type a big integer n:   100
Type a smaller value of n: 0
Type a smaller value of n: 0
Goodbye
venus>
```

Answer:

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

int main() {
   int n, smaller;
   cout << "Type a big integer n:   ";
   cin >> n;
   for (int i = 1; i <= 4; i++) {
      cout << "Type a smaller value of n: ";
      cin >> smaller;
      if (smaller >= n) {
         cout << "Goodbye" << endl;
         return 0;
      }
      n = smaller;
   }
   return 0;
}
```

Problem 2

Write C++ statements to carry out the following tasks. Do not write complete programs, just give a few lines of C++ code. Most answers need no more than two lines. No solution can use more than four lines. Assume that the following variables have been declared.

```
int x, y, age;  double z;  string name, a, b;
```

(a) Print a prompt and then read values from the screen for variables x and y (in this order).

Answer:

```
cout << "Enter integers x and y: ";
cin >> x >> y;
```

(b) If x is negative, assign *name* as *Freddy* otherwise as *Jack*

Answer:

```
if (x < 0) name = "Freddy";
else name = "Jack";
```

(c) Print *name* a total of y times on different output lines.

Answer:

```
for (int c = 1; c <= y; c++) cout << name << endl;
```

(d) Until x and y are not equal, repeatedly make the user enter a new value for y

Answer:

```
while (x == y) {
   cout << "Enter a different value of y: ";
   cin >> y;
}
```

(e) Print the last digit of the larger of x and y

Answer:

```
if (x > y) cout << x % 10;
else cout << y % 10;
```

Problem 3   Consider the following C++ program. Suppose that a user runs the program and enters 10 and then 4 as input.

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

int main() {
    int  a, b, c;
    cout << "Enter two\npositive integers\n";              // line (a)
    cin >> a >> b;

    if ((a > b) && (b > 4)) cout << a - b << endl;         // line (b)
    else cout << b - a << endl;

    c = a + b;
    for (int n = (c / 2); n <= (c * 2); n += 3) cout << n;  // line (c)
    cout << endl;

    cout << c + b / a << endl;                             // line (d)

    cout << (c + b) % a << endl;                           // line (e)

    return 0;
}
```

(a) What is the output at line (a)?

Answer:

```
Enter two
positive integers
```

(b) What is the output from the instruction beginning at line (b)?

Answer:

```
-6
```

(c) What is the output at line (c)?

Answer:

```
710131619222528
```

(d) What is the output at line (d)?

Answer:

```
14
```

(e) What is the output at line (e)?

Answer:

```
8
```

Problem 4   Write a complete C++ program that asks the user for a number n and prints n upward diagonal stripes (each with height n and width n) in a horizontal sequence.

For example, if the user specified 4 for n, the program would print as follows:

```
    *    *    *    *
  *    *    *    *
*    *    *    *
*    *    *    *
```

(Each stripe should begin in the column after the previous one ends. Do not try to check whether the user input is legal or sensible.)

Answer:

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

int main() {
    int n;
    cout << "Enter the number n: ";
    cin >> n;

    for (int r = n; r >= 1; r--) {
        for (int stripe = 1; stripe <= n; stripe++) {
            for (int c = 1; c <=n; c++) {
                if (r == c) cout << "*";
                else cout << " ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

QUEENS COLLEGE   Department of Computer Science
CSCI 111   First Midterm Exam   Spring 2018   03.14.18
Solutions
09.05am – 09.55am, Wednesday, March 14, 2018

Problem 1   Write a complete C++ program that asks the user to type a big integer n. It should then ask 5 times for the user to type a bigger value of n and then thank the user. However, if the user ever enters a value that is not bigger it should immediately exit with no message.

Partial credit will be given for programs that perform some of the required steps but excessively long or complicated programs will lose credit.

Examples of two sample runs of the program:

```
venus> ./a.out
Type a big integer n:   100
Type a bigger value of n: 200          venus> ./a.out
Type a bigger value of n: 300          Type a big integer n:   100
Type a bigger value of n: 400          Type a bigger value of n: 200
Type a bigger value of n: 500          Type a bigger value of n: 150
Type a bigger value of n: 600          venus>
Thank you
venus>
```

Answer:

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

int main() {
   int n, bigger;
   cout << "Type a big integer n:   ";
   cin >> n;
   for (int i = 1; i <= 5; i++) {
      cout << "Type a bigger value of n: ";
      cin >> bigger;
      if (bigger <= n) return 0;
      n = bigger;
   }
   cout << "Thank you\n";
   return 0;
}
```

Problem 2

Write C++ statements to carry out the following tasks. Do not write complete programs, just give a few lines of C++ code. Most answers need no more than two lines. No solution can use more than four lines. Assume that the following variables have been declared.

```
int x, y, age;  double z;  string name, a, b;
```

(a) Print a prompt and read the value of *name* and of *age* from the screen (in this order).

Answer:

```
cout << "Enter name and age: ";
cin >> name >> age;
```

(b) Until *age* is between 10 and 110, repeatedly make the user enter a new value for *age*

Answer:

```
while ((age < 10) || (age > 110)) {
    cout <<"Enter your real age: ";
    cin >> age;
}
```

(c) Print the tens and units digits of *age* (on two lines, in this order):

Answer:

```
cout << (age / 10) % 10 << endl << age % 10 << endl;
```

(d) if the user's *name* is Freddy set *x* to 4 otherwise set *x* to 5

Answer:

```
if (name == "Freddy") x = 4;
else x = 5;
```

(e) Print the user's *name* a total of *x* times separated by spaces

Answer:

```
for (int c = 1; c <= x; c++) cout << name << " ";
```

Problem 3   Consider the following C++ program. Suppose that a user runs the program and enters 6 and then 5 as input.

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

int main() {
    int  a, b, c;
    cout << "Enter two\npositive integers\n";            // line (a)
    cin >> a >> b;

    if ((a > b) && (b > 4)) cout << a - b << endl;        // line (b)
    else cout << b - a << endl;

    c = a + b;
    for (int n = (c / 2); n <= (c * 2); n += 3) cout << n;   // line (c)
    cout << endl;

    cout << c + b / a << endl;                            // line (d)

    cout << (c + b) % a << endl;                          // line (e)

    return 0;
}
```

(a) What is the output at line (a)?

Answer:

```
Enter two
positive integers
```

(b) What is the output from the instruction beginning at line (b)?

Answer:

```
1
```

(c) What is the output at line (c)?

Answer:

```
5811141720
```

(d) What is the output at line (d)?

Answer:

```
11
```

(e) What is the output at line (e)?

Answer:

```
4
```

Problem 4   Write a complete C++ program that asks the user for an odd number n and prints n large X patterns (each with height n and width n) in a horizontal sequence.

For example, if the user specified 5 for n, the program would print as follows:

```
*    **    **    **    **    *
 * *   * *   * *   * *   * *
  *     *     *     *     *
 * *   * *   * *   * *   * *
*    **    **    **    **    *
```

(Each X should begin in the column after the previous one ends. Do not try to check whether the user input is legal or sensible.)

Answer:

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

int main() {
    int n;
    cout << "Enter an odd number n: ";
    cin >> n;

    for (int r = 1; r <= n; r++) {
        for (int cross = 1; cross <= n; cross++) {
            for (int c = 1; c <= n; c++) {
                if ((r == c) || ((r + c) == (n + 1))) cout << "*";
                else cout << " ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

Problem 1   Write a complete C++ program that asks the user 5 questions. The first question asks the user to think of a multiple of 1, the second question asks for a multiple of 2, and so on. However, if the user ever enters a value that is not a multiple as required the program should immediately say *Wrong* and terminate.

Partial credit will be given for programs that perform some of the required steps but excessively long or complicated programs will lose credit.

Examples of two sample runs of the program:

```
venus> ./a.out
Think of a multiple of 1: 4
Think of a multiple of 2: 4
Think of a multiple of 3: 6
Think of a multiple of 4: 4
Think of a multiple of 5: 5
venus>
```

```
venus> ./a.out
Think of a multiple of 1: 4
Think of a multiple of 2: 4
Think of a multiple of 3: 4
Wrong
venus>
```

Answer:

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

int main() {
   int n;
   for (int i = 1; i <= 5; i++) {
      cout << "Think of a multiple of " << i << ": ";
      cin >> n;
      if ((n % i) != 0) {
         cout << "Wrong" << endl;
         return 0;
      }
   }
   return 0;
}
```

Problem 2

Write C++ statements to carry out the following tasks. Do not write complete programs, just give a few lines of C++ code. Most answers need no more than two lines. No solution can use more than four lines. Assume that the following variables have been declared.

```
int x, y, age;  double z;  string name, a, b;
```

(a) Print a prompt and read values from the screen for y and x (in this order).

Answer:

```
cout << "Enter integers y and x: ";
cin >> y >> x;
```

(b) If y is an even number , assign *name* as Freddy otherwise as Jack

Answer:

```
if ((y % 2) == 0) name = "Freddy";
else name = "Jack";
```

(c) Print *name* a total of x times on one line, separted by spaces

Answer:

```
for (int c = 1; c <= x; c++) cout << name << " ";
```

(d) Until x and y are not equal, repeatedly subtract 1 from x and divide y by 2

Answer:

```
while (x == y) {
   x -= 1;
   y = y / 2;
}
```

(e) Print the last digit of the larger of x and y.

Answer:

```
if (x > y) cout << x % 10;
else cout << y % 10;
```

Problem 3   Consider the following C++ program. Suppose that a user runs the program and enters 10 and then 4 as input.

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

int main() {
    int  a, b, c;
    cout << "Enter two\npositive integers\n";            // line (a)
    cin >> a >> b;

    if ((a > b) && (b > 2)) cout << a - b << endl;       // line (b)
    else cout << b - a << endl;

    c = a + b;
    for (int n = (c / 2); n <= (c * 2); n *= 2) cout << n;  // line (c)
    cout << endl;

    cout << c + b / a << endl;                            // line (d)

    cout << (c + b) % a << endl;                          // line (e)

    return 0;
}
```

(a) What is the output at line (a)?

Answer:

```
Enter two
positive integers
```

(b) What is the output from the instruction beginning at line (b)?

Answer:

6

(c) What is the output at line (c)?

Answer:

71428

(d) What is the output at line (d)?

Answer:

14

(e) What is the output at line (e)?

Answer:

8

Problem 4   Write a complete C++ program that asks the user for a number n and prints n squares made of $*$ symbols each with an upward diagonal stripe made of O symbols. Each square has height n and width n and the squares form a horizontal sequence.

For example, if the user specified 4 for n, the program would print as follows:

```
***O ***O ***O ***O
**O* **O* **O* **O*
*O** *O** *O** *O**
O*** O*** O*** O***
```

(Between each pair of squares leave a gap of one blank column.)

Answer:

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

int main() {
    int n;
    cout << "Enter the number n: ";
    cin >> n;

    for (int r = n; r >= 1; r--) {
        for (int square = 1; square <= n; square++) {
            for (int c = 1; c <=n; c++) {
                if (r == c) cout << "O";
                else cout << "*";
            }
            cout << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Problem 1   Write a complete C++ program that asks the user 4 questions. The first question asks the user to think of a number that ends in 1, the second question asks for a number ending in 2, and so on. If the user correctly answers all 4 questions the program should print a message *Well done*. However, if the user ever enters a value with the wrong last digit the program should immediately terminate.

Partial credit will be given for programs that perform some of the required steps but excessively long or complicated programs will lose credit.

Examples of two sample runs of the program:

```
venus> ./a.out
Think of a number that ends in 1: 11
Think of a number that ends in 2: 12
Think of a number that ends in 3: 13
Think of a number that ends in 4: 14
Well done
venus>
```

```
venus> ./a.out
Think of a number that ends in 1: 11
Think of a number that ends in 2: 10
venus>
```

Answer:

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

int main() {
   int n;
   for (int i = 1; i <= 4; i++) {
      cout << "Think of a number that ends in " << i << ": ";
      cin >> n;
      if ((n % 10) != i) return 0;
   }
   cout << "Well done" << endl;
   return 0;
}
```

Problem 2

Write C++ statements to carry out the following tasks. Do not write complete programs, just give a few lines of C++ code. Most answers need no more than two lines. No solution can use more than four lines. Assume that the following variables have been declared.

```
int x, y, age;  double z;  string name, a, b;
```

(a) Print a prompt and read the value of *age* and *name* from the screen (in this order).

Answer:

```
cout << "Enter age and name: ";
cin >> age >> name;
```

(b) Until *age* is between 10 and 120, repeatedly make the user enter a new value for *age*

Answer:

```
while ((age < 10) || (age > 120)) {
    cout <<"Enter your real age: ";
    cin >> age;
}
```

(c) Set *x* as the tens digit and *y* as the units digit of *age*

Answer:

```
x = (age / 10) % 10;
y = age % 10;
```

(d) if the user's *name* is Freddy set *x* to *y* otherwise set *y* to *x*

Answer:

```
if (name == "Freddy") x = y;
else y = x;
```

(e) Print a 2 digit number whose first digit is *x* and second digit is *y*

Answer:

```
cout << x << y;
```

Problem 3   Consider the following C++ program. Suppose that a user runs the program and enters 4 and then 7 as input.

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

int main() {
    int  a, b, c;
    cout << "Enter two\npositive integers\n";              // line (a)
    cin >> a >> b;

    if ((a > b) && (b > 2)) cout << a - b << endl;         // line (b)
    else cout << b - a << endl;

    c = a + b;
    for (int n = (c / 2); n <= (c * 2); n *= 2) cout << n;  // line (c)
    cout << endl;

    cout << c + b / a << endl;                              // line (d)

    cout << (c + b) % a << endl;                            // line (e)

    return 0;
}
```

(a) What is the output at line (a)?

Answer:

```
Enter two
positive integers
```

(b) What is the output from the instruction beginning at line (b)?

Answer:

```
3
```

(c) What is the output at line (c)?

Answer:

```
51020
```

(d) What is the output at line (d)?

Answer:

```
12
```

(e) What is the output at line (e)?

Answer:

```
2
```

Problem 4   Write a complete C++ program that asks the user for a number n and prints n squares made of *
symbols each with an downward diagonal stripe made of Z symbols. Each square has height n and width n and the
squares form a horizontal sequence.

For example, if the user specified 4 for n, the program would print as follows:

```
Z*** Z*** Z*** Z***
*Z** *Z** *Z** *Z**
**Z* **Z* **Z* **Z*
***Z ***Z ***Z ***Z
```

(Between each pair of squares leave a gap of one blank column.)

Answer:

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

int main() {
    int n;
    cout << "Enter the number n: ";
    cin >> n;

    for (int r = 1; r <= n; r++) {
        for (int square = 1; square <= n; square++) {
            for (int c = 1; c <=n; c++) {
                if (r == c) cout << "Z";
                else cout << "*";
            }
            cout << " ";
        }
        cout << endl;
    }
    return 0;
}
```