Problem 1 Write a complete C++ program that asks a user to enter their day and month of birth. If the user's birthday is March 14^{th}, the program wishes the user a Happy Birthday, otherwise it just says Hello. For example, the program could run as follows:

```
What is your day and month of birth:  14 March
Happy Birthday.
```

```
#include <iostream>
using namespace std;
int main() {
   int d;
   string month;
   cout << "Enter your day and month of birth: ";
   cin >> d >> month;
   if (d == 14 && month == "March")
      cout << "Happy Birthday." << endl;
   else cout << "Hello" << endl;
   return 0;
}
```

Problem 2 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print to the screen the message (the word Hello repeats 10 times):

```
Hello Hello Hello Hello Hello Hello Hello Hello Hello Hello
```

```
for (int c = 1; c <= 10; c++) cout << "Hello ";
```

(b) Read an integer from the user and print the integer without its last digit. (For example if the user enters 19683, the program would print 1968.)

```
int n;
cin >> n;
cout << n / 10;
```

(c) Print the square root of 19683 to the output screen:

```
cout << sqrt(19683);
```

(d) Ask the user enter a name. If the user says Freddy, force the user to keep entering a name until something else is received.

```
string name;
cout << "Who are you: ";
cin >> name;
while (name == "Freddy") {
   cout << "No!  Who are you: ";
   cin >> name;
}
```

(e) Print a random number between 1000 and 9999 to the screen.

```
cout << rand() % 9000 + 1000;
```

Problem 3 Consider the following C++ program. Write exactly what output is produced in response to the given user inputs.

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

int main(){
      int n, m;
      cout << "Please two integers: ";
      cin >> m >> n;

      if (n == 0 && m == 0) cout << n << endl;;
      if (n == 0 || m == 0) exit(1);
      if (n < 0 && m < 0) cout << " Negative" << endl;
      else {
         if (n < m) cout << n << endl;
      }
      if (m > 7) cout << " 7" << n << endl;
      return 0;
}
```

(i) The user enters: 0 0

```
0
```

(ii) The user enters: 0 10

(iii) The user enters: -10 -10

```
 Negative
```

(iv) The user enters: 10 -10

```
-10
 7-10
```

(v) The user enters: 10 10

```
 710
```

Problem 4 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an integer that is at least 3.

2. The program reads a value x entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of x has been entered.

3. The program prints a picture with x columns. The picture should display a left pointing arrow pattern.

For example, if the user enters 4 for x the program should print the following picture.

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

Answer:

```
#include <iostream>
using namespace std;
int main() {
  int x;
  cout << "Enter an integer that is at least 3: ";
  cin >> x;
  while (x < 3) {
    cout << "Must be at least 3.  Try again: ";
    cin >> x;
  }
  int rows = 2 * x - 1;

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

Problem 5 Write a complete C++ program that does the following.

1. It asks the user to enter their favorite positive integer.

2. The program prints the square root of that integer.

Here is an example of how the program should work:

```
Enter your favorite positive integer:  25
It has square root: 5.0
```

Answer:

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

int main() {
   int n;
   cout << "Enter your favorite positive integer:  ";
   cin >> n;
   cout << "It has square root: " << sqrt(n) << endl;
   return 0;
}
```

Problem 6 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print to the screen the message:

```
2 + 2 = 4
```

```
cout << "2 + 2 = 4";
```

(b) Print all the numbers from 1 to 1000 to the screen (one number per line).

```
for (int c = 1; c <= 1000; c++)
   cout << c << endl;
```

(c) Ask the user enter a multiple of 3. If the user gives an incorrect response force the user to keep entering a number until a multiple of 3 is received.

```
int n;
cout << "Enter a multiple of 3: ";
cin >> n;
while ( n % 3 != 0 ) {
   cout << "Wrong.  Enter a multiple of 3: ";
   cin >> n;
}
```

(d) Print 10 random numbers each between 10 and 20 to the output screen:

```
for (int c = 1; c <= 10; c++) {
   cout << rand() % 11 + 10 << endl;
}
```

(e) Read an integer greater than 2 from the user, then print its largest factor. (For this problem, a factor of x is a number f with $1 \leq f < x$ that divides into x without remainder.)

```
int n;
cout << "Enter an integer greater than 2: ";
cin >> n;
factor = n - 1;
while ( n % factor > 0) factor--;
cout << factor;
```

Problem 7 Consider the following C++ program. Write exactly what output is produced in response to the given user inputs.

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

int main(){
      int n;  string name;
      cout << "Please enter your name and an integer: ";
      cin >> name >> n;

      if (n == 0 && name == "Freddy") cout << name << endl;;
      if (n == 0 || name == "Freddy") exit(1);
      if (n < 0) cout << " Negative" << endl;
      else {
         cout << " name " << name << " name " << endl;
      }
      if (n > 7) cout << " 7 " << endl;
      return 0;
}
```

(i) The user enters: Freddy 0

```
Freddy
```

(ii) The user enters: Freddy 10

(iii) The user enters: Fred -10

```
 Negative
```

(iv) The user enters: Fred 5

```
 name Fred name
```

(v) The user enters: Fred 10

```
 name Fred name
 7
```

Problem 8 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

The program prints a table with 100 lines of output. On output line number x the program should list the numbers from x to x^2 together with their sum.

For example, the first 4 lines of output read as follows:

```
1 the sum is 1
2 3 4 the sum is 9
3 4 5 6 7 8 9 the sum is 42
4 5 6 7 8 9 10 11 12 13 14 15 16 the sum is 130
```

Answer:

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

int main() {
  for (int x = 1; x <= 100; x++) {
    int sum = 0;
    for (int c = x; c <= x * x; c++) {
      cout << c << " ";
      sum = sum + c;
    }
    cout << "the sum is " << sum << endl;
  }
  return 0;
}
```

Problem 9 Write a complete C++ program that does the following.

1. It asks the user to enter the number of quarters, dimes, nickels and cents that they are carrying.

2. The program then reports the total amount of change that the user has.

Here is an example of how the program should work:

```
How many quarters do you have?  7
How many dimes do you have?     2
How many nickels do you have?   3
How many cents do you have ?    6
That makes 216 cents in change.
```

Answer:

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

```
int main () {
        int q, d, n, p;

        cout << "How many quarters do you have? ";
        cin >> q;

        cout << "How many dimes do you have? ";
        cin >> d;

        cout << "How many nickels do you have? ";
        cin >> n;

        cout << "How many pennies do you have? ";
        cin >> p;

        cout << "That makes "
            << ((q * 25) + (d * 10) + (n * 5) + p) << " in change.";
        cout << endl;

        return 0;
} //main
```

Problem 10 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print a random number between -1 and -9 to the output screen:

Answer:

```
int r = rand() % 9 + 1;
cout << -r;
```

(b) Print (to the output screen) the sum of the square roots of the numbers $1, 2, 3, 4, 5$ and 6.

Answer:

```
double sum = 0;
for (int s = 1; s <= 6; s++)
      sum += sqrt(s);
cout << sum;
```

(c) Ask the user to enter the word "Hello". Force the user to keep entering a new word until an input equal to "Hello" is received.

Answer:

```
string input = "";
while (input != "Hello") {
   cout << "Please enter the word 'Hello': ";
   cin >> input;
}
```

(d) Print twelve random negative numbers.

Answer:

```
for (int n = 1; n <= 12; n++) {
        int r = rand();
        if (r > 0) r = -r;
        cout << r;
}
```

(e) Print the largest integer whose square root is less than 1729.

Answer:

```
int n = 1;
while (sqrt(n) < 1729) n++;
cout << n - 1;
```

Problem 11 Consider the following C++ program. Write exactly what output is produced in response to the given user inputs.

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

int main(){
      int n, m;  string name;
      cout << "Please enter two integers followed by your name: ";
      cin >> m >> n >> name;

      if(n == 0) exit(1);
      if(m >= n) cout << name;
      if(m % n == 1) cout << name << name;
      else while (n > 7) {
        cout << n;
        n = n - m;
      }
      cout << endl;
      return 0;
}
```

(i) The user enters: 2 2 Freddy

Answer:

```
Freddy
```

(ii) The user enters: 0 7 007

Answer:

(iii) The user enters: 1 10 X

Answer:

```
XX
```

(iv) The user enters: 1 2 3

Answer:

```
33
```

(v) The user enters: 11 111 Freddy

Answer:

```
1111008978675645342312
```

Problem 12 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an odd positive integer.

2. The program reads a value n entered by the user. If the value is not legal, the program terminates.

3. The program prints an $n \times n$ grid displaying a large letter X. The left half of the X should be made with the character +, the right half should be made with the character x and the very center should be a $*$.

For example, if the user enters 7 for n the program should print the following picture.

```
+     x
 +   x
  + x
   *
  + x
 +   x
+     x
```

Answer:

```
#include <iostream>
using namespace std;
int main ()
{
        int n;
        cout << "Please enter an odd positive integer ";
        cin >> n;

        if (n < 0 || n % 2 != 1) exit (1);

        int middle = (n / 2) + 1;
        for (int r = 1; r <= n; r++) {
                for (int c = 1; c <= n; c++) {
                        if (r == middle && c == middle)
                                cout << "*";
                        else if ((r == c || r + c == n + 1) && c < middle)
                                cout << "+";
                        else if ((r == c || r + c == n + 1) && c > middle)
                                cout << "x";
                        else
                                cout << " ";
                } //for
                cout << endl;
        } //for

        return 0;
} //main
```

Problem 13 Write a complete C++ program that does the following.

1. It asks the user to enter their age (which is assumed to be a positive integer).

2. The program should print the word *Hello* once for each year of the user's age.

Here is an example of how the program should work:

```
Enter your age:  5
Hello Hello Hello Hello Hello
```

Answer:

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

int main() {
   int age;
   cout << "Enter your age:   ";
   cin >> age;
   for (int x = 1; x <= age; x++)
```

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

Problem 14 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print a random number that has 3 digits to the output screen:

Answer:

```
cout << rand()%900 + 100 << endl;
```

(b) Print (to the output screen) the smallest integer whose square root is larger than 3141.5926:

Answer:

```
int k = (int) (3141.5926 * 3141.5926);
cout << k + 1 << endl;
```

(c) Ask the user to type a password and then to type it again. Print *Error* if the two words are different.

Answer:

```
string pw1, pw2;
cout << "Enter a password twice ";
cin >> pw1 >> pw2;
if (pw1 != pw2) cout << "Error" << endl;
```

(d) Read a positive integer greater than 2 from the user, and print its largest factor. (For this problem a number f is a factor of the number x if $1 \leq f \leq x-1$ and f divides into x without remainder.)

Answer:

```
cout << "Enter a postive integer (greater than 2): ";
cin >> n;
m = n - 1;
while (n % m != 0) m--;
cout << m << endl;
```

(e) Read a name from the user. If necessary, repeatedly ask the user to reenter a name until the user has said *Freddy*.

Answer:

```
string name;
cout << "You are Freddy. What is your name: ";
cin >> name;
while (name != "Freddy") {
   cout << "Wrong! What is your name: ";
   cin >> name;
}
```

Problem 15 Consider the following C++ program. Write exactly what output is produced in response to the given user inputs.

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

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

```
      cout << "Please enter two integers: ";
      cin >> n >> m;

      if (n > m) cout << n % m << endl;
      else {
         for (int r = 1; r < n; r++) {
            for (int c = 1; c < m - n - 1; c++) {
               cout << "*+";
            }
            cout << endl;
            if (n == 10) exit(1);
         }
      }
      return 0;
}
```

(i) The user enters: 10 9

Answer:

```
1
```

(ii) The user enters: 3 7

Answer:

```
*+*+
*+*+
```

(iii) The user enters: 3 15

Answer:

```
*+*+*+*+*+*+*+*+*+*+
*+*+*+*+*+*+*+*+*+*+
```

(iv) The user enters: 10 15

Answer:

```
*+*+*+
```

(v) The user enters: -1 5

Answer:

Problem 16 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer.
2. If the input is illegal, the program should terminate.
3. The program prints the digits of the number in reverse order (separated by spaces) and then gives their sum.

For example, if the user enters 19683 the program should print the following output.

```
3 8 6 9 1 sum to 27
```

Answer:

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

```
int main() {
   int n;
   cout << "Enter a positive integer: ";
   cin >> n;
   if (n <= 0) exit(1);

   int digit, sum = 0;
   while (n > 0) {
      digit = n % 10;
      n = n / 10;
      sum += digit;
      cout << digit << " ";
   }
   cout << "sum to " << sum << endl;

   return 0;
}
```

Problem 17 Write a complete C++ program that does the following.

1. It asks the user to enter their age (which is assumed to be a positive integer).

2. If the user is a teenager, the program should print *Hello Teenager* otherwise it should just print *Hello*.

Here is an example of how the program should work:

```
Enter your age:  15
Hello Teenager
```

Answer:

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

int main() {
   int age;
   cout << "Enter your age:   ";
   cin >> age;
   if ((13 <= age) && (age <= 19)) cout << "Hello Teenager" << endl;
   else cout << "Hello" << endl;
   return 0;
}
```

Problem 18 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print 5 random numbers each between 1 and 9 to the output screen:

Answer:

```
   for (int i = 1; i <= 5; i++) cout << rand()%9 + 1 << endl;
```

(b) Print (to the output screen) the square root of 19683:

Answer:

```
   cout << sqrt(19683) << endl;
```

(c) Ask the user enter a positive integer and if the user gives a non-positive response force the user to keep entering a number until a positive input is received.

Answer:

```
cout << "Enter a positive integer: ";
cin >> n;
while (n <= 0) {
   cout << "Wrong! try again: ";
   cin >> n;
}
```

(d) Read an integer greater than 2 from the user, then print its smallest factor. (For this problem, a factor of x is a number f with $2 \leq f \leq x$ that divides into x without remainder.)

Answer:

```
cout << "Enter a postive integer (greater than 2): ";
cin >> n;
int f = 2;
while (n % f != 0) f++;
cout << f << endl;
```

(e) Read a name from the user and exit the program if the name is *Freddy*.

Answer:

```
string name;
cout << "What is your name: ";
cin >> name;
if (name == "Freddy") exit(1);
```

Problem 19 Consider the following C++ program. Write exactly what output is produced in response to the given user inputs.

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

int main(){
      int n;  string name;
      cout << "Please enter your name and an integer: ";
      cin >> name >> n;

      if(n == 0) cout << name;
      if(n >= 100) exit(1);
      if(n % 5 == 1) cout << name << name;
      else while (n > 7) {
        cout << n;
        n = n - 2;
      }
      cout << endl;
      return 0;
}
```

(i) The user enters: Freddy 0

Answer:

```
Freddy
```

(ii) The user enters: 007 6

Answer:

```
007007
```

(iii) The user enters: Fred 10

Answer:

```
108
```

(iv) The user enters: 9 11

Answer:

```
99
```

(v) The user enters: Freddy 111

Answer:

Problem 20 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer.

2. The program reads a value n entered by the user. If the value is not legal, the program terminates.

3. The program prints a table with n lines of output. On output line number x the program should list the numbers from 1 to x together with their sum.

For example, if the user enters 7 for n the program should print the following table.

```
1 the sum is 1
1 2 the sum is 3
1 2 3 the sum is 6
1 2 3 4 the sum is 10
1 2 3 4 5 the sum is 15
1 2 3 4 5 6 the sum is 21
1 2 3 4 5 6 7 the sum is 28
```

Answer:

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

int main() {
   int r, c, n;
   cout << "How many rows of output do you want: ";
   cin >> n;
   if (n <= 0) exit(1);

   for (r = 1; r <= n; r++) {
      int sum = 0;
      for (c = 1; c <= r; c++) {
         cout << c << " ";
         sum = sum + c;
      }
      cout << "the sum is " << sum << endl;
   }

   return 0;
}
```

Problem 21 Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer x.

2. The program reads the number entered by the user. If x is not a positive integer, the program should terminate.

3. The program prints a countdown from x to 1.

Here is an example of how the program should work:

```
Enter a positive integer:  5
5 4 3 2 1
```

Answer:

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

int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) exit(1);
   for (int n = x; n >= 1; n--) cout << n << " ";
   cout << endl;
   return 0;
}
```

Problem 22 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

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

int print1(int x){
      cout << "Odd" << endl;
      return 1;
}

int print2(int x){
      cout << x*x << endl;
      return x;
}

int main(){
      int n;
      cout << "Please enter a positive integer: ";
      cin >> n;

      if(n <= 0){
            cout << "No good!" << endl; exit(1);
      }

      if (n < 10) {
            cout << n % 2 << endl; exit(0);
      }
      if (n > 11) cout << print1(n) << endl;
      if (n % 2 == 1) print2(n);
      else print1(n);
      return 0;
}
```

(i) The user enters: 0

```
No good!
```

(ii) The user enters: 9

```
1
```

(iii) The user enters: 10

```
Odd
```

(iv) The user enters: 11

```
121
```

(v) The user enters: 12

```
Odd
1
Odd
```

Problem 23 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print (to the output screen) the message:
Very Easy Question

```
cout << "Very Easy Question" << endl;
```

(b) Print (to the output screen) the square root of 11:

```
cout << sqrt(11.0) << endl;
```

(c) Make the user enter 6 decimal values and print their product.

```
double x, product = 1.0;
for (int c = 1; c <= 6; c++) {
   cin >> x;
   product *= x;
}
cout << product << endl;
```

(d) The header line for a function *add3* that calculates the sum of three input integer values. (A header line is a title line, or prototype.)

```
int add3(int x, int y, int z)
```

(e) Print the value of a randomly selected integer between 31 and 41. (The program should make a random selection using the function *rand*. Output values of 31 and 41 are allowed.).

```
cout << rand() % 11 + 31 << endl;;
```

Problem 24 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an even positive integer.

2. The program reads a value n entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of n has been entered.

3. The program prints an $n \times n$ pattern of $*$ symbols in the shape of a large letter U.

For example, if the user enters 6 for n the program should print the following picture.

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

Answer:

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

int main() {
   int c, r, n;
   cout << "Enter a positive even integer: ";
   cin >> n;
   while ((n <= 0) || (n % 2 != 0)) {
      cout << "Illegal.  Try again: ";
      cin >> n;
   }
   for (r = 1; r < n; r++) {
     cout << "*";
     for (c = 2; c < n; c++) cout << " ";
     cout << "*" << endl;
   }
   for (c = 1; c <= n; c++) cout << "*";
   cout << endl;
   return 0;
}
```

Problem 25 Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer x.
2. The program reads the number entered by the user. If x is not a positive integer, the program should terminate.
3. The program repeatedly prints the word *Hello* a total of x times.

Here is an example of how the program should work:

```
Enter a positive integer:  3
Hello Hello Hello
```

Answer:

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

int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) exit(1);
   for (int n = 1; n <= x; n++) cout << "Hello ";
   cout << endl;
   return 0;
}
```

Problem 26 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

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

void print1(int x){
      cout << "Odd" << endl;
}

void print2(int x){
      cout << "Even" << endl;
}

int main(){
      int n;
      cout << "Please enter an integer: ";
      cin >> n;

      if(n == 0) cout << "Hello" << endl;
      if(n <= 10) cout << "Goodbye" << endl;
      if(n > 10 && n%2 == 1) print1(n);
      if(n > 10 && n%2 == 0) print2(n);
      if (n < 0) print2(n);
      return 0;
}
```

(i) The user enters: 12

```
Even
```

(ii) The user enters: 11

```
Odd
```

(iii) The user enters: 10

```
Goodbye
```

(iv) The user enters: 0

```
Hello
Goodbye
```

(v) The user enters: -1

```
Goodbye
Even
```

Problem 27 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print (to the output screen) the message:
Easy Question

```
cout << "Easy Question" << endl;
```

(b) Print (to the output screen) a message made from the first 20 integers:
12345678910111213141516171819 20

```
for (int n = 1; n <= 20; n++) cout << n;
cout << endl;
```

(c) Make the user enter 6 decimal values and print their sum.

```
double x, sum = 0.0;
for (int c = 1; c <= 6; c++) {
   cin >> x;
   sum += x;
}
cout << sum << endl;
```

(d) The header line for a function *max3* that calculates the maximum of three input decimal values. (A header line is a title line, or prototype.)

```
double max3(double x, double y, double z)
```

(e) Print the value of a randomly selected teen age. (The program should make a random selection using the function *rand*. A teen age is a number between 13 and 19.).

```
cout << rand() % 7 + 13 << endl;;
```

Problem 28 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an odd positive integer.

2. The program reads a value n entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of n has been entered.

3. The program prints an $n \times n$ pattern of $*$ symbols in the shape of a large letter T.

For example, if the user enters 7 for n the program should print the following picture.

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

Answer:

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

int main() {
   int c, r, n;
   cout << "Enter a positive odd integer: ";
   cin >> n;
   while ((n <= 0) || (n % 2 == 0)) {
      cout << "Illegal.  Try again: ";
      cin >> n;
   }
   for (c = 1; c <= n; c++) cout << "*";
   cout << endl;
   for (r = 2; r <= n; r++) {
     for (c = 1; c <= n / 2; c++) cout << " ";
     cout << "*" << endl;
   }
   return 0;
}
```

Problem 29 Write a complete C++ program that does the following.

1. It asks the user to enter a positive number x.
2. The program reads the number entered by the user. If x is not positive, the program should terminate.
3. The program prints the square root of x.

Here is an example of how the program should work:

```
Enter a positive number:  6.25
The square root is:  2.5
```

Answer:

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

int main() {
   double x;
   cout << "Enter a positive number: ";
   cin >> x;
   if (x <= 0) exit(0);
   cout << "The square root is:  " << sqrt(x) << endl;
   return 0;
}
```

Problem 30 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

```
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl;
      exit(1);
   }
   for (int i = 1; i <= x % 10; i++)
      cout << x << i << ".";
   cout << x/10 << endl;
}
```

(i) The user enters: 0

Answer:

```
Illegal
```

(ii) The user enters: 1

Answer:

```
11.0
```

(iii) The user enters: 11

Answer:

```
111.1
```

(iv) The user enters: 44

Answer:

```
441.442.443.444.4
```

(v) The user enters: 40

Answer:

```
4
```

Problem 31 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print (to the output screen) the message:

2 + 2 = 4

Answer:

```
cout << "2 + 2 = 4" << endl;
```

(b) Read and store a first name, a middle initial, and a last name as entered by the user.

Answer:

```
string first, middle, last;
cin >> first >> middle >> last;
```

(c) Make the user enter 6 integer values and print the product.

Answer:

```
int x, product = 1;
for (int i = 1; i <= 6; i++) {
   cin >> x;
   product *= x;
}
cout << product;
```

(d) Print the message *odd* if the integer variable x stores an odd value, otherwise print the message *even*.

Answer:

```
if ((x % 2) == 1) cout << "odd" << endl;
else cout << "even" << endl;
```

(e) Print the value of a randomly selected two digit integer. (The program should make a random selection using the function *rand*).

Answer:

```
cout << rand() % 90 + 10 << endl;
```

Problem 32 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an odd positive integer.

2. The program reads a value n entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of n has been entered.

3. The program prints an $n \times n$ pattern in the shape of a star. The pattern should appear as a large X printed from copies of the letter X that lies over a large + printed from copies of the character +.

For example, if the user enters 7 for n the program should print the following picture.

```
X  +  X
 X + X
  X+X
+++X+++
  X+X
 X + X
X  +  X
```

Answer:

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

int main() {
   int n;
   cout << "Enter an odd positive integer: ";
   cin >> n;
   while ((n <= 0) || (n % 2 == 0)) {
     cout << "Illegal. Try again: ";
     cin >> n;
   }

   for (int r = 1; r <= n; r++) {
     for (int c = 1; c <= n; c++)
       if (c == r || ((c + r) == (n + 1))) cout << "X";
       else if ((c == (n + 1) / 2) || (r == (n + 1) / 2)) cout << "+";
       else cout << " ";
     cout << endl;
   }
   return 0;
}
```

Problem 33 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(a) Print (to the output screen) the message:

Problem 1(a).

Answer:

```
cout << "Problem 1(a)." << endl;
```

(b) Read and store a name as entered by the user.

Answer:

```
string name;
cout << "Enter your name: ";
cin >> name;
```

(c) Print the value of the larger of two variables x and y each of which has type int. (For example, if x is 0 and y is 3, the larger value 3 is printed.)

Answer:

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

(d) Make the user enter 10 integer values and print the sum.

Answer:

```
cout << "Enter 10 integers: ";
int x, sum = 0;
for (int c = 1; c <= 10; c++) {
   cin >> x;
   sum = sum + x;
}
cout << sum << endl;
```

Problem 34 Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer x.
2. The program reads the number entered by the user. If x is not positive, the program should terminate.
3. The program prints x randomly generated dice rolls.

Here is an example of how the program should work:

```
Enter a positive number:  3
The dice rolled:  4 1 6
```

Answer:

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;

int main() {
   srand(time(0));
   int x;
   cout << "Enter a positive number:";
   cin >> x;
   if (x <= 0) exit(0);
   cout << "The dice rolled:  ";
   for (int c = 1; c <= x; c++)
      cout << rand()%6 + 1 << " ";
   cout << endl;
   return 0;
}
```

Problem 35 The following C++ program applies 5 different functions. Supply title lines (prototypes) for the 5 functions. Do not supply any blocks of code for the functions.

```
int main() {
   int x, c, r;
   x = readData();
   for (c = 0; c < 5; c++) printValues(x, c);
   x = adjust(x + 2);
   r = max3(x, c, 10);
   return fun(x + c, r - c);
}
```

(a)

Answer:

```
   int readData()
```

(b)

Answer:

```
   void printValues(int a, int b)
```

(c)

Answer:

```
   int adjust(int a)
```

(d)

Answer:

```
int max3(int a, int b, int c)
```

(e)

Answer:

```
int fun(int a, int b)
```

Problem 36 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer.

2. The program reads a value x entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of x has been entered.

3. The program prints a triangular display which has the number x on its top row. Each later row is obtained by omitting the last digit from the number on the previous row.

For example, if the user enters 19683 for x the program should print the following picture.

```
19683
1968
196
19
1
```

Answer:

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

int main() {
   int x = 0;
   while (x <= 0) {
     cout << "Enter a positive integer: ";
     cin >> x;
   }
   while (x > 0) {
     cout << x << endl;
     x = x / 10;
   }
   return 0;
}
```

Problem 37 Write a complete C++ program that does the following.

1. It asks the user to enter a positive even integer.

2. The program reads the number entered by the user. If the value is illegal, the program should terminate.

3. The program calculates and prints the square of the number.

Here is an example of how the program should work:

```
Enter a positive even number:  6
The square is 36.
```

Answer:

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

int main() {
```

```
   int n;
   cout << "Enter a positive even integer: ";
   cin >> n;
   if ((n <= 0) || (n % 2 == 1)) exit(1);
   int ans = n * n;
   cout << "The square is " << ans << "." << endl;
   return 0;
}
```

Problem 38 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(i) Print (to the output screen) the message:

Today is March 19, 2008.

Answer:

```
cout << "Today is March 19, 2008." << endl;
```

(ii) Read and store an age entered by the user.

Answer:

```
int age;
cout << "How old are you? ";
cin >> age;
```

(iii) Print the average 2 variables x and y each of which has type int. (For example, if x is 0 and y is 3, the average is 1.5 and a decimal number must be printed.)

Answer:

```
cout << (x + y) / 2.0 << endl;
```

(iv) Make the user repeatedly enter a value for an integer variable x until the value entered is larger than 10.

Answer:

```
int x = 0;
while (x <= 10) {
   cout << "Enter a value of x that is larger than 10: ";
   cin >> x;
}
```

Problem 39 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

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

int fun1(int x) {
   int ans = x / 10;
   return ans;
}
void fun2(int x) {
   cout << x << "* ";
}
void fun3(int x) {
   cout << "fun3 ";
}
int main() {
```

```
   int x;
   cout << "Enter an integer: ";
   cin >> x;
   if (x < 10) {
      cout << "Too small!" << endl; exit(1);
   }
   if (x == 10) fun3(x);
   if (x >= 20) fun2(x);
   if (x <= 20) cout << fun1(x);
   cout << endl;
   return 0;
}
```

(i) The user enters: 5

Answer:

```
Too small!
```

(ii) The user enters: 15

Answer:

```
1
```

(iii) The user enters: 25

Answer:

```
25*
```

(iv) The user enters: 10

Answer:

```
fun3 1
```

(v) The user enters: 20

Answer:

```
20* 2
```

Problem 40

Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an integer that is at least 2.

2. The program reads a value x entered by the user. If the value is not legal, the program repeatedly makes the user type in another value until a legal value of x has been entered. (Note legal means greater than 1.)

3. The program prints a picture with x rows. The first row should show the first x positive integers, the next row the first $x - 1$ positive integers, until eventually the last row shows only the number 1.

For example, if the user enters 5 for x the program should print the following picture.

```
12345
1234
123
12
1
```

Answer:

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

int main() {
   int x;
   cout << "Enter a value greater than 1 for x: ";
   cin >> x;

   while (x <= 1) {
      cout << "Try again: ";
      cin >> x;
   }

   for (int r = x; r >= 1; r--) {
      for (int c = 1; c <= r; c++) cout << c;
      cout << endl;
   }
   return 0;
}
```

Problem 41 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

```
#include <iostream>
using namespace std;
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl;
      exit(1);
   }
   if (x <= 100) {
      cout << x;
   }
   else {
     cout << x/100 << x%10 << endl;
   }
   return 0;
}
```

(i) The user enters: -50

Answer: Illegal

(ii) The user enters: 0

Answer: Illegal

(iii) The user enters: 99

Answer: 99

(iv) The user enters: 456

Answer: 46

(v) The user enters: 4560

Answer: 450

Problem 42 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(i) Print (to the output screen) the message:

Easy!

Answer:

```
cout << "Easy!" << endl;
```

(ii) Read and store a name entered by the user.

Answer:

```
string name;
cout << "Who are you:";
cin >> name;
```

(iii) Print the value of the larger of 2 variables x and y each of which has type double.

Answer:

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

(iv) Print the difference between 2 variables a and b each of which has type int. (The printed difference should not be negative. For example the difference between 4 and 7 is 3, so too is the difference between 7 and 4.)

Answer:

```
if (a > b) cout << a - b;
else cout << b - a;
```

Problem 43 Write a complete C++ program that does the following.

1. It asks the user to enter a positive number.
2. The program reads the number entered by the user. If the value is not positive, the program should terminate.
3. The program calculates and prints the last digit of the entered number.

Here is an example of how the program should work:

```
Enter a positive number:  56
last digit is 6.
```

Answer:

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

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

   if (x <= 0) exit(1);

   cout << "last digit is " << x % 10 << endl;
   return 0;
}
```

Problem 44

Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer value, x.

2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of x has been entered. (Note positive means greater than 0.)

3. The program prints an $x \times x$ rectangle outlined with $*$ symbols.

For example, if the user enters 5 for x the program should print the following pattern.

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

Answer:

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

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

   while (x <= 0) {
      cout << "Enter a positive number: ";
      cin >> x;
   }

   for (int row = 1; row <= x; row ++) {
     for (int col = 1; col <= x; col++)
       if (row == 1 || row == x || col == 1 || col == x) cout << "*";
       else cout << " ";
     cout << endl;
   }
}
```

Problem 45 The following C++ program is supposed to ask a user to enter their name and date of birth. It then greets the user and wishes a happy birthday if it is the user's birthday. The program has a number of errors. Rewrite the program to fix the errors.

```
#Include <iostream>;
#Include <string>;
Using namespace std;
main() {
cout  "Enter your name and the month, day, and year of your birth: "
int name, month, day, year; cin >> name >> day >> month >> year;
cout  "Hello  name"  endl; if (month = 3 || day = 14) {
cout  "Happy birthday"  endl;
}
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
   cout << "Enter your name and the month, day, and year of your birth: ";
   string name;
```

```
   int month, day, year;
   cin >> name >> month >> day >> year;
   cout << "Hello " <<  name << endl;
   if (month == 3 && day == 14)
      cout << "Happy birthday" << endl;
   return 0;
}
```

Problem 46

Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer value, x.

2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of x has been entered. (Note positive means greater than 0.)

3. The program calculates and prints out x^x. (The value of x^x is $x \times x \times x \times \ldots \times x$, a product of x copies of the number x.)

Answer:

```
#include <iostream>
using namespace std;
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;

   while (x <= 0) {
     cout << "Please enter a positive value: ";
     cin >> x;
   }

   int power = 1;
   for (int i = 1; i <= x; i++)
      power = power * x;
   cout << power << endl;
   return 0;
}
```

Problem 47 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

```
#include <iostream>
using namespace std;
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl;
      exit(1);
   }
   while (x > 0) {
      cout << x % 10;
      x = x / 10;
   }
```

```
   cout << x << endl;
   return 0;
}
```

(i) The user enters: -50 **Answer:** Illegal

(ii) The user enters: 7 **Answer:** 70

(iii) The user enters: 467 **Answer:** 7640

(iv) The user enters a positive integer. (Explain how the output is related to the integer that the user enters.)

Answer: The digits of the input number are printed in reverse order, followed by a 0.

Problem 48 The following C++ program is supposed to ask a user to enter three different integers. It then prints the middle value of the three input numbers. The program has several errors. Rewrite the program to fix the errors and arrange the program so that it is easier for a human to read.

```
# <iostream>;
using namespace std;
int main
    {
int x, y, z;
           cout << "Enter three different integers: " endl;
cin >> "x" >> "y" >> "z" endl;
     if ((x > y > z) && (z > y > x)); cout << y;
     if ((y > x > z) && (z > x > y)); cout << x;
     if ((z > y > x) && (x > y > z)); cout << y; return; };
```

Answer:

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

int main() {
   int x, y, z;
   cout << "Enter three different integers: " << endl;
   cin >> x >> y >> z;
   if ( (x > y && y > z) || (z > y && y > x)) cout << y;
   if ( (y > z && z > x) || (x > z && z > y)) cout << z;
   if ( (z > x && x > y) || (y > x && x > z)) cout << x;
   cout << endl;
   return 0;
}
```

Problem 49 Write a complete C++ program that does the following.

1. It asks the user to enter a positive integer value, x.

2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of x has been entered. (Note positive means greater than 0.)

3. The program prints out a triangle with x rows that points downwards. For example, if the user enters 3 for x the program should print:

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

Answer:

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

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

   while (x <= 0) {
      cout << "Enter a positive integer:";
      cin >> x;
   }

   for (int row = x; row >= 1; row--) {   // print  row stars
      for (int star = 1; star <= row; star++)
         cout << "*";
      cout << endl;
   }
   return 0;
}
```

Problem 50 Consider the following C++ program. Write the exact output that is produced in response to the given user inputs.

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

int fun(int a) {
   int b;  b = a / 10; return b;
}

int main() {
   int x, y;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl; exit(1);
   }
   y = fun(x);
   cout << x << y << endl;
   return 0;
}
```

(i) The user enters: -50 **Answer:** Illegal

(ii) The user enters: 7 **Answer:** 70

(iii) The user enters: 467 **Answer:** 46746

Problem 51 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(i) Print (to the output screen) the greeting:

Hello. This is an easy question.

Answer:

```
cout << " Hello.  This is an easy question." << endl;
```

(ii) Get the user to enter their first name, which is to be stored as the variable *name*.

Answer:

```
string name;
cout << " Enter your first name:";
cin >> name;
```

(iii) Print the sum of the numbers from 1 to 1000 onto the screen. (The output should be the value of $1 + 2 + \ldots + 999 + 1000$).

Answer:

```
int sum = 0;
for (int c = 1; c <= 1000; c++)
   sum = sum + c;
cout << sum << endl;
```

(iv) Get the user to enter an integer value. Print the message *POSITIVE* if it is greater than zero, or *NEGATIVE* if it is less than zero. Do not take any action if the user enters zero.

Answer:

```
int x;
cout << "Enter an integer: ";
cin >> x;
if (x > 0) cout << "POSITIVE" << endl;
if (x < 0) cout << "NEGATIVE" << endl;
```

Problem 52 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

```
#include <iostream>
using namespace std;
void multiPrint(int y) {
   for (int i = 1; i <= y; i++)
      cout << y << "!";
   return;
}
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl;
      exit(1);
   }
   if (x > 2) multiPrint(x);
   cout << x << endl;
   return 0;
}
```

(i) The user enters: -50

Answer:

```
Illegal
```

(ii) The user enters: 1

Answer:

```
1
```

(iii) The user enters: 4

Answer:

```
4!4!4!4!4
```

Problem 53 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter their 4 digit id number.

2. The program reads the number entered by the user and stores it as the variable x. If the value of x is not in the range from 1000 to 9999, the program repeatedly makes the user type in another number until a proper id number has been entered.

3. The program calculates and prints out the last digit of the id number.

For example: A typical interaction with a user might be as follows. (The user responses are shown as bold.)

Enter your 4 digit id: **56789**

Illegal, try again: **25**

Illegal, try again: **9995**

The last digit of your id is 5.

Answer:

```
#include <iostream>
using namespace std;
int main() {
   int x;
   cout << "Enter your 4 digit id: ";
   cin >> x;
   while ( x < 1000 || x > 9999) {
      cout << "Illegal, try again: ";
      cin >> x;
   }
   cout << "The last digit of your id is " << x % 10 << endl;
   return 0;
}
```

Problem 54 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(i) Print (to the output screen) the greeting:

Hello. Today is 10/25/2006.

Answer:

```
cout << "Hello.  Today is 10/25/2006." << endl;
```

(ii) Get the user to enter their age, which is to be stored as the variable *age*.

Answer:

```
int age;
cout << "How old are you: ";
cin >> age;
```

(iii) Get the user to enter a positive integer value. Exit if the user enters a non-positive value, otherwise:

Print the message *EVEN* if the value is even, or *ODD* if it is odd.

Answer:

```
int x;
cout << "Enter a positive value: ";
cin >> x;
if (x <= 0) exit(0);
if ((x % 2) == 0) cout << "EVEN" << endl;
else cout << "ODD" << endl;
```

(iv) Get the user to enter a name. If the user enters the name *Freddy*, tell the user to enter a different name and force the user to re-enter a name until it is different from *Freddy*.

Answer:

```
string name;
cout << "What is your name? ";
cin >> name;
while (name == "Freddy") {
   cout << "That name is illegal.  Give another: ";
   cin >> name;
}
```

Problem 55 Consider the following C++ program. Explain what output is produced in response to the given user inputs.

```
int main() {
   int x;
   cout << "Enter a positive integer: ";
   cin >> x;
   if (x <= 0) {
      cout << "Illegal" << endl;
      exit(1);
   }
   if (x < 10)  cout << x--;
   else {
      if ((x % 10) == 0) cout << x / 10;
      cout << x * 10;
   }
   cout << endl;
}
```

(i) The user enters: -50

Answer: Illegal

(ii) The user enters: 0

Answer: Illegal

(iii) The user enters: 9

Answer: 9

(iv) The user enters: 456

Answer: 4560

(v) The user enters: 4560

Answer: 45645600

Problem 56 Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer value, x.

2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of x has been entered. (Note positive means greater than 0.)

3. The program prints a triangular pattern that is x rows high. The characters 0 and 1 are used to print the pattern. Odd numbered rows are printed using a 1 and even numbered rows are printed using a 0.

For example, if the user enters 4 for x the program should print the following pattern with 4 rows.

```
1
00
111
0000
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
   int x;
   cout << "Give a positive integer value of x: ";
   cin >> x;

   while (x <= 0) {
      cout << "Give a POSITIVE value: ";
      cin >> x;
   }

   for (int r = 1; r <= x; r++) {
      for (int c = 1; c <= r; c++)
         cout << r % 2;
      cout << endl;
   }
}
```

Problem 57 Write C++ statements to carry out the following tasks. Do not write complete programs, just give a single line, or a few lines of C++ instructions.

(i) Print (to the output screen) the greeting:

Hello.

Answer:

```
cout << "Hello." << endl;
```

(ii) Get the user to enter their age. Then print out whichever the following messages applies:

You are over 25

You are not over 25

Answer:

```
int age;
cout << "Enter your age: ";
cin >> age;
if (age > 25) cout << "You are over 25";
else cout "You are not over 25";
cout << endl;
```

(iii) Ask the user to enter an even number.

Make the user re-eneter the number as often as is needed until the number is even.

Answer:

```
int number;
cout << "Enter an even number: ";
cin >> number;
while (number % 2 != 0) {
   cout << "Try again: ";
   cin >> number;
}
```

(iv) Print the average value of 3 variables x, y, and z each of which has type double.

Answer:

```
cout << (x + y + z) / 3;
```

(v) Calculate and print the decimal that represents the fraction $\frac{1}{7}$.

Answer:

```
cout << 1.0 / 7;
```

Problem 58 Write a complete C++ program that does the following.

1. It asks the user to enter a number of cents that is betwee 0 and 99.

2. The program reads the number entered by the user. If the value is not in the right range, the program should terminate.

3. The program calculates and prints out the most efficient combination of quarters, nickels, dimes, and pennies that provide the sum entered by the user.

Here is an example of how the program should work:

```
How many cents? 57
quarters: 2
dimes: 0
nickels: 1
pennies: 2
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
   int cents, q, d, n;
   cout << " How many cents? ";
   cin >> cents;
   if (cents < 0 || cents > 99) exit(1);

   q = cents / 25;
   cents = cents % 25;
   d = cents / 10;
   cents = cents % 10;
   n = cents / 5;
   cents = cents % 5;

   cout << "quarters: " << q << endl
        << "dimes: " << d << endl
        << "nickels: " << n << endl
        << "pennies: " << cents << endl;
   return 0;
}
```

Problem 59

Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter a positive integer value, x.

2. The program reads a value entered by the user. If the value is not positive, the program repeatedly makes the user type in another value until a positive value of x has been entered. (Note positive means greater than 0.)

3. The program prints an $x \times x$ square pattern of $*$ symbols in such a way that rows and columns are separated by rows and columns of $-$ symbols.

For example, if the user enters 3 for x the program should print the following pattern (there are 3 rows and 3 columns that contain $*$'s, but there are other rows and columns with only $-$'s).

```
*-*-*
-----
*-*-*
-----
*-*-*
```

Answer:

```
#include <iostream>
using namespace std;
int main() {
   int x, r, c;
   cout << "Enter a positive integer: ";
   cin >> x;
   while (x <= 0) {
      cout << "Try again: ";
      cin >> x;
   }

   for (r = 1; r <= x; r++) {
      for (c = 1; c <= x; c++) {
        cout << "*";
        if (c < x) cout << "-";
        else cout << endl;
      }
      if (r < x) {
         for (c = 1; c <= 2 * x - 1; c++) cout << "-";
         cout << endl;
      }
   }
}
```