Problem 1  (points) Write a complete C++ program that does the following.

1. It asks the user to enter the base and height of a triangle.
2. If either the base or height is less than or equal to 0, the program should immediately terminate.
3. The program calculates and prints the area of the triangle using the formula:

\[ \text{Area} = \frac{1}{2} \text{Base} \times \text{Height} \]

Here is an example of how the program should work:

Enter the base and height of a triangle: 8 10
Area: 40.0

Answer:

#include <iostream>
using namespace std;

int main() {
    double base, height;
    double area;

    cout << "Enter the base and height of a triangle:";
    cin >> base >> height;
    if ((base <= 0) || (height <= 0))
        return 0;
    area = base * height / 2.0;
    cout << "Area: " << area << endl;
    return 0;
}
Problem 2  (points) 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. Assume that the following declarations have been made and the declared variables have been set to have legal values.

```cpp
string firstName, lastName;
int x, y, z;
double pi;
```

(i) Print to the screen the message:

pi is 22/7

Answer:

```cpp
cout << "pi is 22/7" << endl;
```

(ii) Print to the screen the value of \( x + y + z \).

Answer:

```cpp
cout << x + y + z << endl;
```

(iii) Read in the values of \( x, y, \) and \( z \) (in this order).

Answer:

```cpp
cout << "Enter x,y and z: ";
    cin >> x >> y >> z;
```

(iv) If the value of firstName is Freddy, print the message Goodbye. Otherwise print Hello.

Answer:

```cpp
if (firstName == "Freddy") cout << "Goodbye" << endl;
    else cout << "Hello" << endl;
```

(v) Prompt the user to enter a first name and last name and read their response to appropriate variables.

Answer:

```cpp
    cout << "Enter first name and last name:"
    cin >> firstName >> lastName;
```
Consider the following C++ program. The program makes use of a function `last3digits` that returns the number formed by the last 3 digits of its input argument as its result. So for example, `last3digits(12345678)` would be 678.

Make sure to use your own 8-digit CUNY ID number as the number entered as input to the program. It would be a very bad idea to give answers based on another student's ID number!

```cpp
int main() {
    int id, n, x = 10, y = 17, z = 19;

    cout << "Enter your 8-digit CUNY id number: ";
    cin >> id; // assume that the user types YOUR OWN CUNY ID number
    cout << id << endl; // line (a)
    n = last3digits(id);

    cout << n / x << endl; // line (b)
    cout << ((n % x + 2.0) / 100 << endl; // line (c)

    if ((y > z) || (n % x > x)) cout << "Yes\n"; // line (d)
    else cout << "No\n";
    y += 3; z /= 3;
    cout << z << y << n << endl; // line (e)

    return 0;
}
```

(a) What is the output from the instruction beginning on line (a)?
**Answer:** 12345678 students will have a different answer here and other parts will differ accordingly as explained below.

(b) What is the output from the instruction beginning on line (b)?
**Answer:** 67 or the first 2 digits from the last 3 digits of the id number.

(c) What is the output from the instruction beginning on line (c)?
**Answer:** 0.1 or 2 more than the last digit of the id number all divided by 100.

(d) What is the output from the instruction beginning on line (d)?
**Answer:** No

(e) What is the output from the instruction beginning on line (e)?
**Answer:** 620789 or 620 followed by the last 3 digits of the id number.
Problem 4  (points) 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:

```cpp
#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 << "*
```
```
Problem 5  (points) Write the best title lines for the functions that are called by the following main program. Do not supply the blocks for the functions.

```c++
int main() {
    int x = 12, y = 36, w = 91331, z;
    string name="Freddy", address="Queens NY", number = "five";

    // a. The function initial returns the initial eg "F".
    cout << initial(name) << endl;  // (a)

    // b. The function inState reports whether states match. Here it returns false.
    if ( !inState(address, "NJ") ) cout << "Not from New Jersey\n";  // (b)

    // c. The function cutThrees removes all digits of 3 from a number.
    cutThrees( w );  // (c)
    cout << w << endl;  // prints 911

    // d. The function toNumber turns a string to an integer, here 5
    z = toNumber(number);  // (d)

    // e. A mystery function.
    mystery(mystery(inState(address, "NY")));  // (e)

    return 0;
}
```

(a) Title line for initial as called at the line marked (a).
**Answer:** string initial(string x) or char initial(string x)

(b) Title line for inState as called at the line marked (b).
**Answer:** bool inState(string x, string y)

(c) Title line for cutThrees as called at the line marked (c).
**Answer:** void cutThrees(int &x)

(d) Title line for toNumber as called at the line marked (d).
**Answer:** int toNumber(string x)

(e) Title line for mystery as called at the line marked (e).
**Answer:** bool mystery(bool x)
Problem 6  (points) Write blocks (that is the code) for the functions that are called by the following main program. No block requires more than 5 lines of code. Excessively long and complicated blocks will not receive credit. Do not supply the title lines for the functions. These are given to you.

```cpp
int main() {
    srand(time(0));
    int a = 2, b = 1, c = 3, d;
    d = dice();     // (a) a random number between 1 and 6
    printLargest(a, b, c) << endl;    // (b) prints the largest, here 3
    copy(a, b);          // (c) a is changed to a copy of b
    cout << a << " " << b << endl;   // here 1 1 is printed
    cout << average(a, a, b, c, c) << endl; // (d) the average of 5 numbers
                                        // here 2.22 is printed
    multiPrint("Hello", 4);         // (e) print Hello 4 times
                                        // Hello Hello Hello Hello

    return 0;
}
```

(a) Write the block for the function called at the line marked (a). It has title line:

```cpp
int dice ()
```

it uses a standard random number function to return a random number.

**Answer:**

```cpp
{
    return rand() % 6 + 1;
}
```

(b) Write the block for the function called at the line marked (b). It has title line:

```cpp
void printLargest(int x, int y, int z)
```

it prints the largest of 3 parameters.

**Answer:**

```cpp
{
    int answer = x;
    if (y > answer) answer = y;
    if (z > answer) answer = z;
    cout << answer << endl;
}
```

(c) Write the block for the function called at the line marked (c). It has title line:

```cpp
void copy(int &x, int y)
```

it changes parameter x to become what y was originally.

**Answer:**

```cpp
{
    x = y;
}
```

(d) Write the block for the function called at the line marked (d). It has title line:

```cpp
double average(int a, int b, int c, int d, int e)
```

returns the average of the five numbers.

**Answer:**

```cpp
{
    return (a + b + c + d + e) / 5.0;
}
```
(e) Write the block for the function called at the line marked (e). It has title line:
void multiPrint(string s, int n)
it prints n copies of s on a line (separate copies with single spaces).

Answer:
{
    for (int i = 1; i <= n; i++) {
        cout << s << " ";
    }
    cout << endl;
}