Describe the order of operations that will occur in the following C++ expression.
\[ x*y/z+(n/m)-7+m%n-1; \]
Write an equivalent algebraic expression for this.
What is the last operation done?
The first?

Write the following for structure as an equivalent while structure. Assume any variable used has been declared and initialized.
\[
\text{sum} = 0; \\
\text{for}(i=1; \ i<N; \ i++) \\
\quad \text{sum} = \text{sum} + 1; \\
\]
Write a logically equivalent set of instructions that uses do-while.

```
sum = 0;
for(i=1; i<N; i++)
    sum = sum + i;
```

What will be printed by the following loop structure?

```
for(i=1; i<=5; i++)
    for(j=i; j>=1; j--)
        cout << '*';
cout << '\n';
```

What will be printed by the following loop structure? (--j instead of j--)

```
for(i=1; i<=5; i++)
    for(j=i; j>=1; --j)
        cout << '*';
cout << '\n';
```
In the preceding problem, what would happen if the {} are omitted?

for(i=1; i<=5; i++)
    for(j=i; j>= 1; j--)
        cout << '*' ;
    cout << 'n';
Describe exactly what will be printed.

What will be printed by the following?

N=5;
while(--N > 2){
    for(i=1; i<= N; i++)
        cout << '*';
    cout << endl;
}

What will be printed by the following?

N=5;
while(N-- > 2){
    for(i=1; i<= N; i++)
        cout << '*' ;
    cout << endl;
}
Suppose that the variable $N$ has been initialized to some positive integer greater than 1. Write a code segment that will find the first positive integer greater than 1 that divides $N$. The first step here is to determine which looping structure you want to use.

Write a program segment that will allow the user to input numbers until a negative number is input. It should compute the sum and average of all the numbers, not counting the negative number terminating the list. (The negative number in this case is called a sentinel and is used to designate the end of a list.)

How many times will the code in \{ \} be executed? Give the answer in terms of $N$, $M$ and $P$.

```c
for(i=1; i<= N; i++)
  for(j=1; j<= M; j++)
    for(k=1; k<= P; k++)
      \{ \text{code} \}
```
How many times will the code in {} be executed? Give the answer in terms of I.

```
for(I=1; I<= N; I++)
    for(J=1; J<=I; J++)
        {code}
```

What is the rule for creating user-defined identifiers? Describe it completely. Are the following legal user-defined identifiers?

1. _the_Maximum
2. 9_of_spades
3. Suit%4
4. true

Write a function called by random(a,b); that will return an integer in the range from a to b inclusive. Assume that a < b.
Write a code segment that will generate 1000 random double numbers in the range [0,1] and as they are generated count how many are in [0, .2], (.2, .5], (.5, 9], and (.9, 1.0]. Place the sums in S1, S2, S3, and S4 respectively. What can be done with this type of simulation?

What is printed by the following code segment with the given function definition?

```cpp
int A = 5;
int B = 8;
cout << A << "  " << B << endl;
foo(A,B);
cout << A << "  " << B << endl;
---------------
void foo(int C, int &D){
 C++; D++;
cout << C << "  " << D << endl;
return 0;
}
```

What is the purpose of a function prototype? Where are prototypes placed in your program? What does it mean to say that a function behaves like a “black box”? 

---

2/7/2005
switch

What happens if there are no break statements in a switch statement?
What happens if every case has a break?
What happens if none of the designated cases equals the value on which the switch is don?

ax^2 + bx + c = 0 \quad -b \pm \sqrt{b^2 - 4ac} \\
2a

Write a function called by roots(a, b, c, r1, r2) that will compute the roots of the quadratic equation above using the given formula. It should place the roots in r1 and r2.