Given that main is calling a function hyp that will calculate and return the length of the hypotenuse c = square root of (a squared + b squared). Write the code for function hyp using calls to sqrt and pow (or write it your own way).

```
#include <iostream>
#include <cmath>
using namespace std;
double hyp(double , double);

int main()
{
   double a= 4.0;
   double c, b=3.0;
   c = hyp(a, b);
   cout << "C = " << c << endl;
   return 0;
} /* should print 5 for c */


__________ hyp( ________ ____, _______ ____)
{









}
```

What output is produced by the following code?

```
#include <iostream>
using namespace std;
void figureMeOut(int& x, int y, int& z) {
   cout << x << " " << y << " " << z << endl;
   x = 1;
   y = 2;
   z = 3;
   cout << x << " " << y << " " << z << endl;
}

int main( ) {
   int a=10, b=20, c=30;
   figureMeOut(a, b, c);
   cout << a << " " << b << " " << c << endl;
   return 0;
}
```

Write a function, either24(int a[], int n), that returns true if the array contains a 2 next to a 2 or a 4 next to a 4, but not both.

either24({1, 2, 2}, 3) → true

either24({4, 4, 1}, 3) → true

either24({4, 4, 1, 2, 2}, 5) → false

Find the errors in the following program segment. Assume the following declarations and statements:

```
1  int *zPtr; // zPtr will reference array z
2  int number;
3  int z[ 5 ] = { 1, 2, 3, 4, 5 };
4
5  zPtr = z;
6
7  // use pointer to get first value of array
8  number = zPtr;
9
10 // print entire array z
11 for ( int i = 0; i <= 5; i++ ) {
12    cout << zPtr[ i ] << endl;
13 }
```

Fill in the blanks for the following code:

A local zoo wants to keep track of how many pounds of food each of its three monkeys eats each day during a typical week. Write a program that stores this information in a two-dimensional 3 by 5 array, where each row represents a different monkey and each column represents a different day of the week. The program should first have the user input the data for each monkey. Then it should create a report that includes the following information:

-Average amount of food eaten per day by the whole family of monkeys.

-The least amount of food eaten during the week by any one monkey.

-The greatest amount of food eaten during the week by any one monkey.

Input Validation: Do not accept negative numbers for pounds of food eaten.

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

int main()
{
     // Array dimension constants
     _____ int NUM_MONKEYS = 3;
     const int NUM_DAYS = 5;

     // 2-D array to hold the food consumed each weekday for 3
different monkeys
     double monkeyFood[NUM_MONKEYS][________]
          = { { -1.0, -1.0, -1.0, -1.0, -1.0 },
              { -1.0, -1.0, -1.0, -1.0, -1.0 },
              { -1.0, -1.0, -1.0, -1.0, -1.0 } };

     // Counter variables
     int i, j;

     // Variables used to hold and/or calculate the average,
minimum, and maximum values
     double min = -1.0, max = -1.0;
     double total = 0.0, average = 0.0;

     for (i = 0; ___ < NUM_MONKEYS; i++)
     {
          cout << "For monkey " << i + 1 << ", please enter the"
               << "number of pounds of food eaten by the monkey"
               << endl;
```

```
          for (____ = 0; j < NUM_DAYS; j++)
          {
               // Input the amount of food the monkeys for each
               // day
               cout << setw(15) << "For day " << j + 1 << ": ";
               cin >> __________[i][j];
               while (monkeyFood[i][j] < 0)
               {
                    cout << "Your entry for day " << j + 1
                         << " was invalid.  Please try again: ";
                    cin >> monkeyFood[i][j];
               }
               // Calculate the sum
               ________ += monkeyFood[i][j];

               // Check if the entry read in becomes a minimum
and/or maximum
               if (monkeyFood[i][j] < min || min == -1.0)
               {
                    min = monkeyFood[i][____];
               }
               if (monkeyFood[____][j] > max)
               {
                    max = monkeyFood[i][j];
               }
          }
     }

     // Calculate the average
     average = (double) total / (NUM_MONKEYS * NUM_DAYS);

     cout << endl << "The average number of pounds of food eaten
by all of the monkeys is " << average << " pounds." << endl;
     cout << "The minimum amount of pounds eaten by any one
monkey was " << _______ << " pounds." << endl;
     cout << "The maximum amount of pounds eaten by any one
monkey was " << _______ << " pounds." << endl;

     return 0;
}
```