1. How would you test whether \( x \) is the ascii code of an upper-case letter, without using any library functions and without mentioning any numbers, and only one line of code?

   \textit{Answer:} \texttt{if('A' <= x && x <= 'Z')} 

2. What will be printed by the following code?

   \begin{verbatim}
   void main(void)
   { int i;
     for(i=0;i<5;i++)
       printf("%c",'a' + i);
   }
   \end{verbatim}

   \textit{Answer:} abcde

3. How many bytes are required to store the string, "without doubt"

   \textit{Answer:} 14, one for each character plus one for the null terminator.

4. If you have to compute with integers that will go up to a few million, what data type should you use?

   \textit{Answer:} int

5. If you have to compute with money values representing the Federal budget (with numbers into the trillions of dollars, but accuracy to the penny is also needed), what data type should you use?

   \textit{Answer:} long long. \textit{Not doubles because we need perfect accuracy, so we have to represent amounts in pennies.}

6. Suppose that \textit{price} is an array of n doubles giving the price of silver for the years 2000-2008. Exhibit two lines of code that will print a nice two-column table of these prices, with the years in the left-hand column and
the prices in the right-hand column. (Do not worry about making a title, just print the data.) The first line should start with for and the second line with printf. The prices should be printed with the decimal points lined up one in the same column, exactly two digits after the decimal, no leading zeroes except in case the amount is less than one dollar. You may assume the prices are all less than $1000.

for(i=0;i<n;i++)
    printf("\n%d %4.2lf", 2000+i, price[i]);

You could also put the newline at the end of the line instead of the beginning.

7. What will be printed by the following program?

void f(int x)
{ x = x+5;
}
void main(void)
{ int x=5;
  f(x);
  printf("%d",x);
}

Answer: 5 is printed. You don't even have to read the definition of f to answer this.

8. Consider the following definition of a function:

void f(int *x)
{ *x = *x + 1;
}

Show how to correctly call this function, using exactly two lines of code. The first line declares a variable and the second line calls f.

Answer, line 1: int n;
Answer, line 2: f(&n);

9. Declare an array of 100 ints.
Answer: int x[100];

10. Suppose x is an array of integers, and we have just executed this code:

\[\text{for}(i=0; i<10; i++)\]
\[x[i] = i+1;\]

Suppose that \(x[0]\) is stored at address 4500. What is the value of each of the following expressions?

(a) \(x[4500]\)
(b) \&x[0] 4500, same as \(x\).
(c) \(*x\) 1, the value that was placed in \(x[0]\).
(d) \(x[1]\) 2
(e) \&x[1] 4504, four bytes more than the value of \(x\), which is \&x[0].

11. In the following code, identify the
   (a) local variables: \textit{sum, i}. Not \textit{z} since an array is a constant, not a variable. You can’t assign a different value to \textit{z}.
   (b) global variables: \textit{username}
   (c) formal parameters: \textit{x, n}
   (d) actual parameters: \textit{z, 10}
   Please write your answers above, not on the code below.

\begin{verbatim}
char *username = "Jack the Ripper";
double average ( double *x, int n)
{ double sum=0.0; int i;
  for(i=0; i<n; i++)
    sum = sum + x[i];
  return sum/n;
}
void main(void)
{
  double z[10];
  int i;
  for(i=0; i<10; i++)
    z[i] = i;
  printf("\%d",average(z,10));
}
\end{verbatim}
12. If you use a computer program to solve the equation \( x = \cos x \), representing numbers as doubles, which of the following would be the answer you could expect to get? (Circle the letter of the correct answer.)

(a) 0.73908513321516064165531208767387340401341175890076
(b) 0.73908513321516064165531208767387
(c) 0.7390851332151611 This is the answer, since doubles have about 15 digits of accuracy.
(d) 0.739085

13. Correct the following code, in which \texttt{pigLatinWord} is as in your homework assignment.

```c
void main(void)
{
    char *ans; // char ans[128];
    pigLatinWord("Cat", ans);
    printf(ans);
}
```

14. Let \texttt{FormatMoney} be as in your homework assignment. How should it be called to format the amount, one hundred twelve dollars and thirteen cents? Your answer should have four lines matching the following comments:

```c
char ans[128]; // declare a variable for the formatted string
long long x; // declare and initialize a variable for the money
FormatMoney(ans,x); // call the FormatMoney function
printf(ans); // print out the answer to see if it worked
```

15. Correct the following code. Each line of the code except the last has an error or errors. Once corrected, the code should compile and the comment in the last line should be true.

```c
void main(void)
{
    char *x = "cat dog giraffe"; // char x[128] = "cat dog giraffe";
    char *firstToken = strtok(x," "); // char *firstToken = strdup(strtok(x," "));
    char *secondToken = strtok(x,NULL); // char *secondToken = strdup(strtok(NULL," "));
    char *thirdToken = strtok(x,NULL); // char *thirdToken = strdup(strtok(NULL, " "));
    printf("%s %s", firstToken, secondToken); // prints cat dog
    // free(firstToken); free(secondToken); free(thirdToken);
}
```
16. When you computed the Fibonacci numbers in your homework assignment, as the numbers got larger, what happened first?

(a) The Fibonacci numbers exceeded the capacity of a long long so the computed long long answers were complete garbage, while the double answers were still approximately right.

(b) The double answers, while still approximately right, did not have enough precision to be accurate to the nearest integer, while the long long answers were still completely accurate. This happens first, since doubles have only 51 bits of mantissa, while long longs have 64 bits.

17. When you computed the Fibonacci numbers in your homework assignment, as the numbers got larger, what happened eventually?

(a) The Fibonacci numbers exceeded the capacity of a long long so the computed long long answers were complete garbage, while the double answers were still approximately right. Doubles can go up to about $10^{308}$, much larger than the largest long long, which is $2^{63}$ or about $10^{21}$. So (a) is correct.

(b) The Fibonacci numbers exceeded the range of a double so the computed double answers were complete garbage, while the long long answers were still correct.

18. In your homework assignment Palindrome, you had to count lower-case and upper-case versions of the same character as equivalent. Show how that test can be written, by putting some code inside the if in the following:

```
int equivalent(char x, char y)
// return 1 if x and y are either equal, or they are
// the same except one is uppercase and one is lowercase.
{
  if( toupper(x) == toupper(y) )
    return 1;
  return 0;
}
```

19. What will be printed by the following code?

```c
void main(void)
{
  int i;
  int x[100];
  for(i=0;i<100;i++)
  
  x[i] = i;
}
printf("%d",i);  // 100 will be printed, the final value of i.
}

Answer:

20. The code in the previous problem is poorly written, in that the hard-coded number 100 occurs twice, so if later the programmer wants to change it to 200, there will be an opportunity for error (by not changing both occurrences). On this copy of the code, show how to correct that problem. Be sure you write ANSI C, not C++.

#define N 100  // No semicolon! This is not an assignment statement
            // You can't use a variable since in C you can't have a
            // variable array dimension.
void main(void)
{
    int i;
    int x[N];  // You can do this since N is a constant when the compiler gets it
    for(i=0;i<N;i++)
        x[i] = i;
    printf("%d",i);
}

Another correct solution is to change the for-loop to

for(i=0;i<sizeof(x)/sizeof(int);i++)

but while this works in this particular problem, it’s not the right solution in general, as there probably will more more occurrences of the number in question.