1. Break the following command line into tokens. Explain in words what each token in this command line does.

```
g++ -g example.cpp -o example 2> errors.txt
```

2. For each of the following code blocks (the full program is not shown – only the first part of `main()`), if there are no errors write down what the output will be. If there is an error, which might be syntactic, logical, or something which will produce a run-time exception, find the error and correct it.

Explain your reasons:

2a.

```c
int i;
int arr[10];
for (i = 1; i <= 10; i++)
    arr[i] = i*i;
```

2b.

```c
int j;
j = 64;
while (j < 1024) {
j = 2 * j;
    cout << "\n" << j ; }
```

2c.

```c
int *ptr;
int nums[4] = { 4, 6, 8, 3 };
int i;
    ptr = nums;
    for (i = 0; i < 4; i++) {
        cout << "\n" << *ptr ;
        ptr++ ; }
```

3. You are to write a program in which the user enters a list of double-precision positive real numbers ending with the sentinel 0.0. Which of the following can be found *without* using an array? Explain why or why not in each case:

3a. The largest number.

3b. The median of the numbers.

3c. The average of the numbers.

3d. The closest two numbers.

4. The following block of code uses a structure to hold course information and initializes it but the programmer is not sure whether he did it correctly. Correct any lines which have errors.

```c
struct course {
    char dept_name[32];
    char course_name[64];
    int course_number;
    int course_units;
} cs221;
struct course *courseptr = &cs221;
```
strcpy(cs221.dept_name, "Computer Science Department");
strcpy(courseptr->course_name, "Computer Science I");
courseptr->course_number = 221;
(*courseptr).course_units = 5;

5. Write code for the procedure below which accepts a string \texttt{str[\]} and \emph{rotates the letters left} (i.e if the string contains \texttt{"hello"} before the call, it contains \texttt{"elloh"} after). You will need a temporary character variable, why? \textbf{There is no need for \texttt{cin} or \texttt{cout} in this procedure.}

\begin{verbatim}
void rotate_left(char str[\])
\end{verbatim}

6. Consider the following block of code below (in \texttt{main()}):

\begin{verbatim}
int z = 4;
cout << "\n z=" << z; func5(z);
cout << "\n z=" << z; func6(z);
cout << "\n z=" << z; func7(&z);
cout << "\n z=" << z;
\end{verbatim}

The functions are defined as follows:

\begin{verbatim}
func5(int z) {
  z = 5;
  return;
}
func6(int & z) {
  z = 6;
  return;
}
func7(int *z) {
  *z = 7;
  return;
}
\end{verbatim}

What will appear on the screen when the block of code is run?

7. A programmer has written a function \texttt{sum_squares()} which is supposed to find the sum of the squares of an array \texttt{data[]} of \texttt{num} double precision reals. The programmer does \textbf{not} want the function to return any value. The code for the function is below. Write a correct and working prototype for it.

\begin{verbatim}
{ int i;
  sum = 0.0;
  for (i = 0 ; i < num ; i++)
    sum = sum + data[i]*data[i];
  return;
}
\end{verbatim}

8. One programmer says that using a structure to hold information about each of his company’s clients would be simple, efficient and a good idea. A second programmer has told him that since some of the information (e.g. credit rating) is sensitive, he would be better advised to use a class and take the time to declare some fields \texttt{private} and control access through member functions. Which programmer do you think has given the better advice and why?

9. The following program has a scoping error in it and will not compile. Find, explain, and correct the error.

\begin{verbatim}
int main() {
  int sum = 0;
  for (int temp = 1 ; temp <= 10 ; temp++) {
    sum = sum + temp;
  }
  cout << "\n sum=" << sum << " and temp=" << temp;
\end{verbatim}
10. Trace the following block of code and write exactly what appears on the screen when it is run (show spaces, newlines, and everything).

```cpp
int index = 47;
int *intptr;

return(0);
}
```

11. A programmer declares the following class and writes the following member functions:

```cpp
class stuff { 
public:
    stuff(int size=100);
    void load_data(int n, double x[]);
    void print_data(void);
private:
    double *dptr;
    int maxnums;
    int actnums; 
};
stuff::stuff(int size) {
    dptr = new double[size];
    maxnums = size;
    actnums = 0;
    return; }
stuff::load_data(int n, double x[]) {
    int i;
    for (i = 0 ; i < n ; i++)
        dptr[i] = x[i];
    actnums = n;
    return; }
stuff::print_data(void) {
    int i;
    for (i = 0 ; i < actnums ; i++)
        printf(" %8.3f", dptr[i]);
    return; }
```

Write the code for `main` which will

i. create an object `widget` of class `stuff` with `size` equal to 10.

ii. load `widget` with the five data values 3.52, -6.71, 8.85, 0.23, 9.23.

iii. print these data values from `widget`.

How does the data appear on the screen? What are the values in `widget.maxnums` and `widget.actnums`? Can these be accessed directly from `main`? Why or why not?