1. The compiler will not report an error. But, depending on the compiler, it may give you a warning. The program may or may not crash depending on whether you have overwritten important information in memory.

2. What is passed is the base address of the array.

3. \( x = ++a[1]; \) For this one, \( a[1] \) is incremented \textit{before} getting assigned to \( x \). That is, what happens here is \( a[1]++ \) followed by \( x = a[1] \). So, \( x = 2 \), and \( a[1] \) becomes 2.

\( y = a[1]++; \) Here, what happens is: \( y = a[1] \) and then \( a[1]++ \). So, \( y = 2 \) and \( a[1] \) becomes 3;

\( z = a[x++]; \) This is equivalent to \( z = a[x] \) followed by \( x++ \). This means \( z = z[2] = 15 \) and \( x \) becomes 3.

\textbf{The output is:} 3 2 15

4. Doing it by hand: It is faster and you have full control.
   Garbage collector: removes the burden from the programmer.

5. a. 502  
b. 502  
c. 502  
d. 10000  
e. 502

6. Nothing is wrong, the declaration is perfectly fine!

7. There is an error in this structure declaration. The structure \texttt{emp} contains a member \texttt{e} of the same type \texttt{struct emp}. At this stage compiler does not know the size of this structure. In the previous problem (#6 above), the compiler knows how big a pointer is.

8. a) 8  
b) 8