5.4. Give a formula for determining the maximum number of bits required for storage of any value in the integer subrange M..N, where M and N are any two integers such that M < N.

5.8. Figure 5.3 (Also drawn in the class) illustrates two number representations for integers with a run-time type descriptor. One uses extra space to gain speed in arithmetic; the other sacrifices speed for a more compact storage structure. Design two similar representations for your local computer assuming the run-time descriptor requires at most 6 bits. Write the programs necessary for addition, subtraction, multiplication, and division of numbers in these forms. Compare the relative advantages and disadvantages of the two representations.

5.11. Concatenation is a central operation on character strings.
   (a) Assuming the representation of Figure 5.5 (Lecture #5, slide #19) for character strings of variable length with a declared bound, design a concatenation operation CAT1. CAT1 is called with three parameters: A, B, and C. A and B are pointers to the two storage blocks containing the strings to be concatenated, and C is the receiving block that initially contains some other character string. The string composed of the characters of String B concatenated to the characters of String A is to be stored in Block C (with the appropriate descriptor, of course). Blocks A and B are to be unchanged by the operation.
   (b) Strings without a declared bound may also be stored sequentially using the same storage representation with the maximum length deleted from the descriptor. Design an appropriate storage structure assuming that characters may be packed four per word. Then design the concatenation operation CAT2. CAT2 has two parameters, A and B, representing the strings to be concatenated and returns a pointer to a new block of storage containing the concatenated strings. Assume CAT2 calls a function ALLOCATE(N) that returns a pointer to a newly allocated block of N words of storage.
   (c) Design CAT3 – a routine that concatenates strings represented as linked lists in the manner of Figure 5.5.