2. **Evaluate the following, assuming the letters are consecutive characters.**

   a. $\text{ord}(\text{d}) - \text{ord}(\text{A})$
   b. $\text{ord}(\text{A}) - \text{ord}(\text{d})$
   c. $\text{ord}(\text{A}) - \text{ord}(\text{G})$
   d. $\text{ord}(\text{c}) - \text{ord}(\text{A})$
   e. $\text{ord}(\text{Z}) - \text{ord}(\text{A}) + 5$

1. Write a function `upperCase` (your own version of `UPCase`) that returns as its value the uppercase equivalent of its character argument or, if there is none, returns the value of its argument.

   a. \(\text{chr}(\text{ord}(\text{d}) - \text{ord}(\text{A}))\)
   b. \(\text{chr}(\text{ord}(\text{A}) - \text{ord}(\text{d}))\)
   c. \(\text{chr}(\text{ord}(\text{G}) - \text{ord}(\text{A}))\)
   d. \(\text{chr}(\text{ord}(\text{A}) - \text{ord}(\text{c}))\)
   e. \(\text{chr}(\text{ord}(\text{Z}) - \text{ord}(\text{A}) + 5)\)

2. The following type declaration and procedure is supposed to ensure the range of an input number is correct, but it generates range check errors anyway. Why?

   ```
   type Year = 1990..2000:
   procedure Safeyear (var Y: Year):
   begin
     Readln (Y);
     repeat
       write ('Input year 1990-2000:');
       Readln (Y);
     until (Y >= 1990) and (Y <= 2000);
   end;
   ```

1. Write a type declaration for a type `Month` appropriate to represent the parameter `M` (type `Month`). Your procedure should prompt the user to enter a month as a word and return a valid month number.

2. Explain why a compiler cannot determine whether a range check error may occur at run time for a particular assignment statement.