1. Write and test a function `deblank(input argument, output argument)` that takes a string output and a string input argument and returns a copy of the input argument with all blanks removed.
   Input: Hello World
   Output: HelloWorld

2. Write a function fact_calc that takes a string output argument and an integer input argument n and returns a string showing the calculation of n! . For example, if the value supplied for n were 6, the string returned would be "6! = 6 x 5 x 4 x 3 x 2 x 1 = 720". Write a program that repeatedly prompts the user for an integer between 0 and 9, calls fact_calc and outputs the resulting string in a box of asterisks of the right size to surround the result. If the user inputs an invalid value, the program should display an error message and reprompt for valid input. Input of the sentinel -1 should cause the input loop to exit.
   Sample run:
   Enter an integer between 0 and 9 or -1 to quit => 6
   ****
   * 6! = 6 x 5 x 4 x 3 x 2 x 1 = 720 *
   ****
   Enter an integer between 0 and 9 or -1 to quit => 12
   Invalid entry
   Enter an integer between 0 and 9 or -1 to quit => 0
   ****
   * 0! = 1 *
   ****
   Enter an integer between 0 and 9 or -1 to quit => -1

3. Write a function `hydroxide` in C that checks if a given string ends with the substring `OH`. The function should return `1` if the string ends with `OH` and `0` otherwise.

   Test the function with the following strings:

   - `KOH`
   - H_2O_2
   - `NaCl`
   - `NaOH`
   - $C_9H_8O_4$
   - `MgOH`

   The program should print the result for each string, indicating whether it ends with `OH`.

4. Write a program in C that takes a list of nouns and generates their plural forms based on the following rules:

1. If a noun ends with **`y`**, replace **`y`** with **`ies`**.
2. If a noun ends with **`s`**, **`ch`**, or **`sh`**, append **`es`**.
3. For all other cases, append **`s`**.

The program should print each noun along with its plural form.

Test the program with the following nouns:

`Chair`, `dairy`, `Boss`, `circus`, `fly`, `Dog`, `church`, `clue`, `dish`