*Star Trek* was a popular science-fiction television series that aired from 1966 to 1969.  Later, it inspired several movies and sequel television series that continue to be produced today.  An important plot element of the show is the ability to use spacecraft to travel from Earth to other solar systems.  Because space is so vast, to travel to other stars in a reasonable amount of time requires us to travel faster than the speed of light!  This is where the concept of warp speed comes in.  Speeds faster than light are measured by a "warp" number, which ranges from 1 to 9.  Warp speed is defined as follows:  warp speed w is equivalent to $w^3$ (w-cubed) times the speed of light.  For example, Warp 5 means a speed that is 125 times the speed of light.

You will write a program that tells the user how long it can take to arrive at a certain set of destinations given a certain warp speed.  First, your program should read a text file called `stars.txt`.  Each line of this file contains the name of a star, and its distance from Earth measured in light years.  The file is formatted so that the first 15 characters of each line are allocated for the star name, and beyond the 15$^{th}$ character is a real number for the distance.  For example, the file stars.txt may look like this:

```
Altair         16.75
Antares        609.2
Betelgeuse     641.8
Castor         51.49
```

After reading the input file, you should interactively ask the user for the desired light speed, which will be a real number from 1 to 9.  You need to perform <u>error checking</u> on this input.  There are two kinds of input errors to watch out for:

- The user enters something that can't be interpreted as a float, such as a word.  In this case, print an error message "Sorry, you need to enter a real number.  Try again."  And then give the user another chance to enter the value correctly.
- The user enters a number less than 1 or greater than 9.  Here, print an error message "Sorry, your number is out of range.  Try again."  Give the user the opportunity to re-enter the value.

In order to handle the error checking in this manner, you need to use both a while loop and a `try`/`except`.

Once your program has obtained a legitimate warp number, it should calculate the amount of time to reach each of the stars listed in stars.txt.  When interpreting distance in light years, assume that 1 year equals exactly 365.2425 days.  Round each time to the nearest hour.  Print the number of days and hours as whole numbers.  You do not need to worry about printing the words day and hour in the singular.  Plural output is fine for this assignment.

In the output, you need to allocate 15 characters for the name of the star, plus a space to separate it from the numerical output.  Therefore, the first digit of the number of days should be the 17$^{th}$ character on the line.  Print the star destinations in the same order that they appeared in stars.txt.

Here is some example I/O for the above set of stars.

```
What is the desired warp speed?  (1-9)  seven
Sorry, you need to enter a real number.  Try again.

What is the desired warp speed?  (1-9)  17
Sorry, the number is out of range.  Try again.

What is the desired warp speed?  (1-9)  7

Altair          17 days 20 hours
Antares         648 days 17 hours
Betelgeuse      683 days 10 hours
Castor          54 days 20 hours
```