An interesting property of the male honeybee is that it only has a single parent. This situation arises because a female honeybee is able to lay eggs that may or may not have been fertilized by a male honeybee. If a female honeybee lays an egg that has not been fertilized by a male, the result will always produce a male honeybee. If, on the other hand, a female honeybee lays an egg that is fertilized by a male, the result will always produce a female honeybee. A female honeybee always has two parents, whereas a male honeybee only has a single (female) parent.

A local apiarist (beekeeper) wants you to build a computer program that will allow her to determine the number of ancestors of a given male honeybee in a given ancestral generation. She inherited an apiary from her great-great-grandparents (who were meticulous record keepers) and wants to begin performing genetic analysis of certain inherited attributes among the honeybees within her hives.

Figure 1 depicts the ancestry of a single male honeybee to three ancestral generations. Each vertex in the figure represents a honeybee. Each vertex is labeled with the gender and generation number of the honeybee. For example, the vertex labeled $M_0$ in Figure 1 represents a honeybee from the current (or $0^{th}$) generation, and each vertex labeled $F_3$ in Figure 1 represents a third-generation female ancestor of the honeybee $M_0$.

Build a C++ program that will allow the apiarist to determine the number of ancestors of a male honeybee in a given ancestral generation. Your program shall allow the apiarist to input the number of the generation that she wants to determine the size of. The apiarist has specified that she will only want to determine the number of ancestors for at most 25 ancestral generations. Your program shall validate user input, so if the user accidently enters an ancestral generation number less than 0 or greater than 25, your program shall print a message indicating the error and immediately terminate.
Figure 1: Ancestry of Single Male Honeybee to Four Generations

Figure 2 is an example of how your program should operate for the inputs specified. Please make sure to follow this example as closely as possible, because the apiarist is quite particular about what she wants.

```
bash 4-1$ ./hw1
Enter ancestral generation number: 2
Ancestral generation 2 contains 2 honeybees

bash 4-1$ ./hw1
Enter ancestral generation number: 0
Ancestral generation 0 contains 1 honeybee

bash 4-1$ ./hw1
Enter ancestral generation number: 26
Error: input is less than 0 or greater than 25

bash 4-1$ ./hw1
Enter ancestral generation number: 13
Ancestral generation 13 contains 377 honeybees
```

Figure 2: Sample Program Run for Four Inputs
Please place your program in a file named “hw2.cpp” (without quotes.) When you are building your program, keep in mind that the apiarist will likely ask you to modify, extend, and/or refine your program in the future. Be sure to analyze the problem before sitting down to “type it up.” Time that is spent in analysis is quite often time that is not spent debugging the program. And debugging the program is usually more troublesome than analyzing the problem.

When you have finished your program, please use escheckin to submit it to your instructor.