Problem 1 (A simple calculator)
Design a program that acts as a rudimentary integer calculator. It should prompt a user for a number, then an operator \{+,-,*,^\} and then another number, all in one input line. The result of \textit{number1 \{operator\} number2} should then be printed to the screen. Only division operations that result in remainders should have their remainders displayed. The program should test for invalid operators. In such a case it should display a suitable error message, followed by a prompt to re-enter the operator.

\textbf{Hint:} Use the if- and case-statements, see p.83 for info on the use of scanf.

Example output:

```
Enter calculation: 20 \& 7
Error: Invalid operator, please retry
Enter calculation: 20 / 7
Answer: 20 / 7 = 2
Remainder: 6
```

Press any key to continue

when no remainder results:

```
Enter calculation: 20 * 2
Answer: 20 * 2 = 40
```

Press any key to continue

\textbf{Problem 2} (Equivalency of for and while loops)
Design a program that finds all the prime numbers smaller or equal to a number \textit{n}, using a for-loop. The number \textit{n} should be entered by the user and the prime numbers should be printed to the screen. Repeat the problem, this time replacing the for-loop with an equivalent while-loop. Which is more intuitive?

\textbf{Hint:} Large numbers can take very long to process...

Example output:

```
Enter number: 8
Prime numbers smaller or equal to 8:
2,
3,
5,
7,
That’s all!
```

Press any key to continue