1. Write a program to request and validate a password from your user.
A valid password will be any two digit integer, both digits of which are even.
Give the user **three chances** to enter a correct password.

- At each incorrect attempt, print “Invalid password. Try again”
- If the password entered is correct print “Correct! You may access the system.” then terminate the program.
- If after three incorrect attempts the password is still incorrect, print “Too many invalid attempts. Please try again later”

2. Write a program that asks the user for a positive integer n where the right-most digit is not a zero.
Construct and output the integer whose digits are the reverse of those in n.
For example, if n has the value 123, then you need to **construct the integer 321 (three hundred twenty one)**. Note you are not just printing the digits of the original number backwards, you are actually constructing the new integer.

3. An integer greater than one is called a prime number if its only positive divisors (factors) are one and itself.
So, 5 is a prime number because its only divisors are 1 and 5, but 6 is not a prime (it’s called a composite number) because its divisible by 1,2,3,and 6.
Write a program that inputs a positive integer n and determines if n is prime or composite.

4. Generate and print all prime numbers between 1 and 100, five numbers/line.

5. Input an integer n, and then print out each digit, one per line.

6. Input an integer n, and then print out the sum of the digits of n.

7. Input an integer n, and then print out the divisors of n, one per line. For example if n=6, your program would print:

1
2
3
6

8. A number, n, is called perfect if it is the sum of its divisors, including 1 but excluding n. For example 6 is a perfect number because 6=1+2+3.
Print out all the perfect number between 1 and 1000, 5 per line.