1. (4.1) Write a program that asks the user for an integer. Calculate (and then print) the sum of all square numbers up to and including the user's number.

   For example,

   - if user_number = 3, the result should be 14 since $1^2 + 2^2 + 3^2 = 14$.
   - if user_number = 8, the result should be $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2 = 204$.

2. (4.2) Write a program that repeatedly asks the user for integers until a negative integer is given.
   Report back the largest **even** number the user entered (not including the negative number).
   If the user didn't enter any even numbers report back -1.

   For example,

```
Enter a number: 6
Enter a number: 5
Enter a number: 3    Enter a number: 4      Enter a number: 3
Enter a number: 4    Enter a number: 3      Enter a number: 9
Enter a number: 8    Enter a number: 2      Enter a number: 7
Enter a number: 5    Enter a number: 1      Enter a number: 5
Enter a number: 6    Enter a number: 0      Enter a number: -4
Enter a number: -2   Enter a number: -1     largest = -1
largest = 8          largest = 6
```

3. (4.3) You are the newest rug fashion designer on the scene, but you're running out of ideas. Write a program that will help you design rugs. The program should ask for a width, a length, and pattern, and then create a rug consisting of that pattern and dimensions.

   For example,

```
Enter a width: 3              Enter a width: 16
Enter a length: 5             Enter a length: 5
Enter a pattern: $            Enter a pattern: @

Your rug is:                  Your rug is:
$$$                           @@@@@@@@@@@@@@@@
$$$                           @@@@@@@@@@@@@@@@
$$$                           @@@@@@@@@@@@@@@@
$$$                           @@@@@@@@@@@@@@@@
$$$                           @@@@@@@@@@@@@@@@
```

1. (4.1) Using a loop, write code to calculate the sum of all odd numbers between 50 and 517. Print the result.

2. (4.2) Write a program that repeatedly asks the user for integers until a negative integer is given.
   Report back the largest **even** number the user entered (not including the negative number).
   If the user didn't enter any even numbers report back −1.

   For example,

   ```
   Enter a number: 6
   Enter a number: 5
   Enter a number: 4
   Enter a number: 3
   Enter a number: 2
   Enter a number: 1
   Enter a number: 0
   Enter a number: -1
   largest = 6
   ```

   ```
   Enter a number: 3
   Enter a number: 4
   Enter a number: 8
   Enter a number: 5
   Enter a number: 6
   Enter a number: -2
   largest = 8
   ```

   ```
   Enter a number: 3
   Enter a number: 9
   Enter a number: 7
   Enter a number: 5
   Enter a number: -4
   largest = -1
   ```

3. (4.3) Ask the user for an integer height, and then build a triangle of asterisks (*) with that height.

   For example,

   ```
   Enter a height: 5

   Here is a triangle of height 5:
   *
   **
   ***
   ****
   *****
   ```

   ```
   Enter a height: 12

   Here is a triangle of height 12:
   *
   **
   ***
   ****
   *****
   ******
   *******
   ********
   *********
   **********
   ***********
   ************
   ```

1. (4.1) Write code that asks the user for an integer and then prints each number that is a factor of the input.

   For example,

   ```
   Enter a number: 12
   1 2 3 4 6 12
   ```

   ```
   Enter a number: 17
   1 17
   ```

   ```
   Enter a number: 36
   1 2 3 4 6 9 12 18 36
   ```

2. (4.2) Write a program that repeatedly asks the user for integers until a negative integer is given. The program should keep track of the sum of the numbers and print the sum at the end (not including the negative number).

   For example,

   ```
   Enter an integer: 7
   Enter an integer: 10
   Enter an integer: 3
   Enter an integer: -4
   20
   ```

   ```
   Enter an integer: 1
   Enter an integer: 2
   Enter an integer: 3
   Enter an integer: 4
   Enter an integer: 5
   Enter an integer: -1
   15
   ```

3. (4.3) You are the newest rug fashion designer on the scene, but you're running out of ideas. Write a program that will help you design rugs. The program should ask for a width, a length, and pattern, and then create a rug consisting of that pattern and dimensions.

   For example,

   ```
   Enter a width: 3
   Enter a length: 5
   Enter a pattern: $

   Your rug is:
   $$$
   $$$
   $$$
   $$$
   $$$
   ```

   ```
   Enter a width: 16
   Enter a length: 5
   Enter a pattern: @

   Your rug is:
   @@@@@@@@@@@@@@@@
   @@@@@@@@@@@@@@@@
   @@@@@@@@@@@@@@@@
   @@@@@@@@@@@@@@@@
   @@@@@@@@@@@@@@@@
   ```

1. (4.1) Write a program that asks the user for a word and then, <u>using a loop</u>, prints every other letter of the word starting with the second letter.

   Examples:

   - if user_word = "counterattack", the result should be "oneatc"
   - if user_word = "banana sunday", the result should be "aaasna"

2. (4.2) Given a positive integer n, the following rules will always create a sequence that ends with 1, called Hailstone Sequence:

   (a) If n is even, divide by 2
   (b) If n is odd, multiply by 3 and add 1 (i.e. $3n + 1$)
   (c) Continue until n is 1

   Write a program that prints the hailstone sequence starting at $n = 25$.

3. (4.3) Ask the user for an integer height, and then build a triangle of asterisks (*) with that height.

   For example,

```
Enter a height: 5

Here is a triangle of height 5:
*
**
***
****
*****
```

```
Enter a height: 12

Here is a triangle of height 12:
*
**
***
****
*****
******
*******
********
*********
**********
***********
************
```