Let's address each question and task step by step:

1. A shorter way to express range (0, 5, 1) is simply range (5).
2. range (x, 10*x, x) represents the sequence [2, 4, 6, 8, 10, 12, 14, 16, 18].
3. range (x, x + y) represents the sequence [2, 3, 4, 5, 6].
4. It's not possible to represent the sequence 1, -1, 2, -2, 3, -3, 4, -4 using a Python range expression because range generates arithmetic progressions, and this sequence is not an arithmetic progression. You'd need to use a different approach to generate this sequence.
5. It's not possible to represent the sequence 1, -2, 3, -4, 5, -6, ..., n-1 using a Python range expression because range generates arithmetic progressions, and this sequence alternates between positive and negative numbers. You'd need to use a different approach to generate this sequence.

Now, let's provide the exact sequences for each of the given range expressions:

a. range (5) generates the sequence [0, 1, 2, 3, 4].
b. range (5, 10) generates the sequence [5, 6, 7, 8, 9].
c. range (5, 20, 3) generates the sequence [5, 8, 11, 14, 17].
d. range (20, 5, -1) generates the sequence [20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6].
e. range (20, 5, -3) generates the sequence [20, 17, 14, 11, 8].
f. range (10, 5) doesn't generate any sequence as the start is greater than or equal to the stop.
g. range (0) doesn't generate any sequence as it's an empty range.
h. range (10, 101, 10) generates the sequence [10, 20, 30, 40, 50, 60, 70, 80, 90, 100].
I. range (10, -1, -1) generates the sequence [10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0].
j. range (-3, 4) generates the sequence [-3, -2, -1, 0, 1, 2, 3].
k. range (0, 10, 1) generates the sequence [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

Now, let's provide equivalent range expressions for the given integer sequences:

a. range (1, 6)
b. range (5, 0, -1)
c. range (5, 31, 5)
d. range (30, 4, -5)
e. range (-3, 4)
f. range (3, -4, -1)
g. range (-50, -9, 10)
h. range (0)

For the last task, writing a Python for loop to output even numbers between 80 and 100 in descending order:

for num in range (100, 79, -1):

```
    if num % 2 == 0:
        print(num)
```

This loop iterates over the range from 100 to 80 (inclusive) in steps of -1 (descending order), and for each number, it checks if it's even (num % 2 == 0) and prints it if it is.

```
========== RESTART: /Users/krishahemani/Desktop/xyz.py ==========
100
98
96
94
92
90
88
86
84
82
80
```

```
for num in range(100, 79, -1):
    if num % 2 == 0:
        print(num)
```