Cashbox

Write a program that reads in an amount to be paid (int) and an amount received (int) from user. The program calculates the change, prints it, and terminates.

If the amount received is too small, a corresponding error message is printed, and the user input is repeated.

If the amount to be paid and/or the amount received is less than 0, the input is incorrect. In that case, the program should also print a message and repeat the user input.

Example usage of the program:

```
To pay:-100
Negative payment is invalid.
To pay:50
Received:-35
Negative received amount is invalid.
To pay:80
Received:25
You did not pay enough.
To pay:75
Received:100
Your change is: 25
```

Sum Up

Write a program that reads in two integers n1 and n2 from the user and outputs all integers from n1 to n2 (inclusively).

If n1 or n2 (or both) are smaller or equal to 0, the program should print a corresponding message. If both numbers are greater than 0, but n2 is less than n1, an error message should also be printed.

Example usage of the program:

```
n1: 2
n2: 8
2 3 4 5 6 7 8
```

```
n1: 8
n2: 2
n2 needs to be > n1
```

```
n1: -1
n2: -3
n1 and n2 need to be > 0
```

Order of Numbers

Write a program that repeatedly asks users to enter three numbers (integer) and outputs whether the numbers entered are in ascending order, descending order, or in neither.

The program then prints "ascending", "descending" or "no specific order".

Additionally, if the entered numbers are not ordered ascending or descending, it checks if all numbers are even (divisible by 2) or odd.

Example usage of the program:

```
n1: 1
n2: 2
n3: 3
numbers are ascending
```

```
n1: 3
n2: 2
n3: 1
numbers are descending
```

```
n1: 0
n2: 4
n3: 2
no specific order, but all even
```

```
n1: 1
n2: 5
n3: 3
no specific order, but all odd
```

```
n1: 1
n2: 5
n3: 2
no specific order
```

Power of Ten

Write a program that asks the user for an at max three-digit integer. The program then shows the decomposition of the number into powers of ten.

If the number provided is negative, print "number cannot be negative", if the number has more than three digits, print "number has more than 3 digits".

Hint: Use floor division (//) and modulo (%) to decompose the numbers. E.g.: 25//10 = 2; 25%10 = 5

Example usage of the program:

```
Enter a max 3 digit number: -10
number cannot be negative
Enter a max 3 digit number: 2457
number has more than 3 digits
Enter a max 3 digit number: 8
8 = 8 * 1
Enter a max 3 digit number: 16
16 = 1 * 10 + 6 * 1
Enter a max 3 digit number: 256
256 = 2 * 100 + 5 * 10 + 6 * 1
```