



1. Grid.

Write a program `gridxo.py` that reads three integer command line arguments (n , r , c) and draws an $n \times n$ grid with the character `x`, except for the row with index r and the column with index c that will contain the character `o`. For example,

```
> python gridxo.py 5 2 3
x x x o x
x x x o x
o o o o o
x x x o x
x x x o x
```



produces a 5×5 table where the row with index 2 (third row) and column with index 3 (fourth column) contain `o`.

2. Without vowels.

Write a function that takes in a string and returns a version of this string that does not contain any vowels, according to the specification below. Vowels are uppercase and lowercase 'a', 'e', 'i', 'o', 'u'.

```
# Returns a version of the string str without vowels and whose
# characters appear in the same order they appear in str
def without_vowels(str):
    # Your code here
```

For example, if $s = \text{'This is great!'}$ then `without_vowels(s)` will return `'Ths s grt!'`. If $s = \text{'a'}$ then the empty string will be returned.

Write a `vowels.py` program that prints the values returned by the function above when called with the strings: 'AUB', 'cmcs 200', and 'Exam 1!'. There is no need to read command line arguments.

3. Local maxima.

Write a program `lmax.py` that reads a sequence of positive integers from the command line and finds and prints all local maxima in the sequence. A value is a local maximum if it is larger than both the preceding and succeeding values in the sequence. Neither the first nor the last values of the sequence can be local maxima.

Your program must include:

- a function `strings2ints(ls)` that takes in a list of strings, and constructs and returns a list of integers; and
- a function `lmax(li)` that takes in a list of integers, and constructs and returns a list of all local maxima.

Your program should call these functions as needed, and print how many local maxima there are and their values. For example, given the sequence 4 2 1 6 8 13 11 6 12 5, the program should print that there are 2 local maxima 13 and 12 (since they are both larger than their neighbors: 8 and 11, and 6 and 5).

```
> python lmax.py 4 2 1 6 8 13 11 6 12 5
2 local maxima: 13 12
```

Insight Club

4. Token.

When you are ready to submit, get from your proctor your individualized 4-character token string and write a one line program `token.py` that contains a single statement of the form `token = 'AB12'` which assigns to the variable `token` the value you get from the proctor.

Submission. Zip the four `.py` files above (`gridxo.py`, `vowels.py`, `lmax.py`, and `token.py`) in a single archive file `exam1_netid` where `netid` is your AUBnet user name, and submit to Moodle.