1. Create a function that takes the width, height and character and returns a picture frame as a 2D list.

Examples

```
get_frame(4, 5, "#")     [
  ["####"],
  ["#  #"],
  ["#  #"],
  ["#  #"],
  ["####"]
]
# Frame is 4 characters wide and 5 characters tall.


get_frame(10, 3, "*")     [
  ["**********"],
  ["*        *"],
  ["**********"]
]
# Frame is 10 characters and wide and 3 characters tall.


get_frame(2, 5, "0")     "invalid"
# Frame's width is not more than 2.
```

2. Write three functions:

  1. boolean_and
  2. boolean_or
  3. boolean_xor

These functions should evaluate a list of True and False values, starting from the leftmost element and evaluating pairwise.

Examples

```
boolean_and([True, True, False, True])     False
# [True, True, False, True] => [True, False, True] => [False, True] => False

boolean_or([True, True, False, False])     True
# [True, True, False, True] => [True, False, False] => [True, False] => True

boolean_xor([True, True, False, False])     False
# [True, True, False, False] => [False, False, False] => [False, False] =>
False
```

3. Create a function that creates a box based on dimension n.

Examples

```
make_box(5)    [
  "#####",
  "#   #",
  "#   #",
  "#   #",
  "#####"
]

make_box(3)    [
  "###",
  "# #",
  "###"
]

make_box(2)    [
  "##",
  "##"
]

make_box(1)    [
  "#"
]
```

4. Given a common phrase, return False if any individual word in the phrase contains duplicate letters. Return True otherwise.

Examples

```
no_duplicate_letters("Fortune favours the bold.")    True

no_duplicate_letters("You can lead a horse to water, but you can't make him drink.")    True

no_duplicate_letters("Look before you leap.")    False
# Duplicate letters in "Look" and "before".

no_duplicate_letters("An apple a day keeps the doctor away.")    False
# Duplicate letters in "apple", "keeps", "doctor", and "away".
```

5. Write a regular expression that will match the states that voted yes to President Trump's impeachment. You must use RegEx positive lookahead.

Example

txt = "Texas = no, California = yes, Florida = yes, Michigan = no"
pattern = "yourregularexpressionhere"

re.findall(pattern, txt) ["California", "Florida"]