

```
def all_positive(nums):
    for num in nums:
        if num <= 0:
            return False
    return True
```

7. Write the exact output of each snippet, one printed line per line.

A

```
nums = [3, 4, 5]
for num in nums:
    total = 0
    total = total + num
print(total)
```

5

B

```
for i in range(3):
    row = ""
    for j in range(i + 1):
        row = row + str(j)
    print(row)
```

0
01
012

C

```
word = "hello"
for i in range(len(word)):
    if i % 2 == 0:
        print(word[i])
```

h
l
o

D

```
words = ["hi", "cat", "a"]
sizes = []
for w in words:
    sizes.append(len(w))
print(sizes)
```

[2, 3, 1]

8. Write a function `above_average(nums)` that returns a new list of the numbers in `nums` that are greater than the average of `nums`, in their original order. If `nums` is empty, it returns `[]`. Otherwise, it returns the new list, which may be empty. For example:

```
>>> above_average([2, 9, 4, 7])
[9, 7]
>>> above_average([5, 5])
[]
>>> above_average([])
[]
```

```
def above_average(nums):
    if len(nums) == 0:
        return []
    avg = sum(nums) / len(nums)
    result = []
    for num in nums:
        if num > avg:
            result.append(num)
    return result
```

9. Write a function `index_of_max(nums)` that returns the index of the largest number in `nums`, without using `max()`. Assume `nums` is not empty. If one number is larger than all the others, it returns that number's index. Otherwise, it returns the index of the first of the tied largest numbers. For example:

```
>>> index_of_max([3, 8, 1])
1
>>> index_of_max([3, 8, 1, 8])
1
>>> index_of_max([-4, -2, -9])
1
```

```
def index_of_max(nums):
    best = 0
    for i in range(len(nums)):
        if nums[i] > nums[best]:
            best = i
    return best
```