

CSCI 134 Midterm Practice Exam

This is a 90-minute closed-book examination. All work should be your own. **Simple and concise solutions** will receive the best score. If you get stuck on a question, **move on** and come back to it later. You may use the back of any page as scratch paper. Good luck!

Please make sure to include your anonymous ID on **every page** of the exam.

Name: _____ **SAMPLE ANSWERS** _____

Anonymous ID: ____ **024** ____

This exam has 6 questions and 13 total pages. The point breakdown is below.

Question	Points	Score
Question 1	12	
Question 2	10	
Question 3	7	
Question 4	14	
Question 5	9	
Question 6	18	
TOTAL	70	

Please sign the honor code statement below.

I have neither given nor received unauthorized aid on this exam.

Signed: _____ **CS134 Fall 2024** _____

Question 1: Sequence Operations (12 pts, 2 pts each)

Fill in the blanks in the **Assignment** column in the following table such that variable `alice` produces `bob`. The assignment expression may use indexing, slicing, arithmetic or sequence operators, and any built-in functions in Python. The simplest assignments will receive full credit. The first row is filled in as an example.

Value of <code>alice</code>	Value of <code>bob</code>	Assignment
<code>[1, 2, 3, 4, 5]</code>	<code>3</code>	<code>bob = alice[2]</code>
<code>['C', 'S', '1', '3', '4']</code>	<code>['4']</code>	<code>bob = [alice[-1]]</code> OR <code>[alice[4]]</code>
<code>[14, 13, 12, 11]</code>	<code>4</code>	<code>bob = len(alice)</code>
<code>('a', 'e', 'i', 'o', 'u')</code>	<code>False</code>	<code>bob = 'x' in alice</code>
<code>list(range(5))</code>	<code>3</code>	<code>bob = alice[3]</code>
<code>'a_l_i_c_e'</code>	<code>'ice'</code>	<code>bob = alice[4::2]</code>
<code>'3.14'</code>	<code>3.14</code>	<code>bob = float(alice)</code>

Question 2: Short Answer (10 pts total, 2pts each)

Answer the following questions **in the space provided** using a **maximum of two sentences**. Longer answers will not receive full credit. This question has subparts A-E.

2A. Fill in the blanks below.

Which type of loops are used to iterate over a predetermined sequence of items?

_____ **for loops** _____

Which type of loops are used when we don't know in advance when the loop will stop?

_____ **while loops** _____

2B. Fill in the blanks below.

A data type is _____ **immutable** _____ if its value cannot be changed after it has been created. An example of such a data type in Python is ____ **int/str/float/boolean** ____.

2C. Consider the following assignment operations:

```
pi = [3, 1, 4, 1, 5, 9]
pi_digits = set(pi)
pie = pi
pi[5] = pi[5] - 1
```

Fill in the following blanks with True or False.

```
>>> pie == pi                >>> len(pi) == len(pi_digits)
```

_____ **True** _____

_____ **False** _____

2D. Name two data types in Python that you can iterate over in a for loop.

1. _____ **str** _____
2. _____ **list** _____

2E. Suppose we have the following definition:

```
scientists = ['Zhou', 'Neumann', 'Pixel', 'Oak', 'Lovelace']
```

What does 'Lovelace' in scientists return?

_____ **True** _____

What does ['Lovelace'] in scientists return?

_____ **False** _____

Question 3: Hundreds Digit (7 pts)

Complete the following function that takes a positive integer as its argument, and returns the “hundreds digit” (as an `int`) – this is the third digit from the right. You can assume that the argument `num` is a positive integer. If `num` is less than 100, then the function should return 0.

```
def hundreds_digit(num):  
    """...  
    >>> hundreds_digit(12345)  
    3  
    >>> hundreds_digit(901)  
    9  
    >>> hundreds_digit(24)  
    0  
    """  
  
    num = num // 100    # 100's digit is now in 1's place  
    num = num % 10      # num is now just the digit in the 1's place  
    return num
```

Question 4: Code Comprehension (14 pts total)

Answer the questions in the right column of the tables below.

4A. (4 points) Consider the following function definitions (on the left).

<pre>def outcome(num): if num in [1, 3, 5]: num = num + 1 elif num in [2, 4, 6]: num = num - 1 if num % 2 == 0: print('oddly even') else: print('evenly odd') def update(num): return num % 7 def big_hero(num): outcome(update(num)) outcome(update(num + 1))</pre>	<p>What is printed when <code>big_hero(6)</code> is called?</p> <p>evenly odd oddly even</p>
--	--

4B. (4 points) Consider the following function definitions (on the left).

<pre>def append(element, l): return l + [element] def prepend(element, l): return [element] + l def big_hero(num): nums = list(range(num, num+5)) append(11, nums) nums = prepend(5, nums) print(nums[0]) print(nums[-1])</pre>	<p>What is printed when <code>big_hero(6)</code> is called?</p> <p>5 10</p>
---	---

4C. (2 points) Consider the following code (on the left).

<pre>word = "stressed" word = word[::-1] word = word[:3] + word[4:7] print(word)</pre>	<p>What is printed when the code on the left is executed?</p> <p>desert</p>
--	---

4D. (2 points) Consider the following code (on the left).

<pre>x = 0 y = 11 z = 5 while x < y: z = z + 10 x = x + 2 y = y - 3 print(z)</pre>	<p>What is printed when the code on the left is executed?</p> <p>35</p>
---	---

4E. (2 points) Consider the following code (on the left).

<pre>x = [1, 2] y = [3, 4] image = [x, y] image[0][1] = 5 print(image[0]) print(x)</pre>	<p>What is printed when the code on the left is executed?</p> <p>[1, 5] [1, 5]</p>
--	--

Question 5: Simplification (9 pts total, 3 pts each)

Rewrite the proceeding snippets of code (subparts A-E) in a **simpler** manner. The **most concise solutions** will receive full credit. Answers should fit in the space provided.

5A. Assume `a` and `b` are ints. Rewrite this function using as few if statements as possible.

```
def either_is_negative(a,b):  
    if a < 0:  
        return True  
    elif b < 0:  
        return True  
    else:  
        return False
```

```
def either_is_negative(a,b):  
    return a < 0 or b < 0
```

5B. Assume `date` is an `int` and `day` is a `str`. Rewrite the following using a single if statement.

```
if day == "Friday":  
    if date == 13 or day == "Thursday":  
        print("Spooky")
```

```
if day == "Friday" and date == 13:  
    print("Spooky")
```


5C. Assume n is a positive integer. Rewrite this code to compute the same value for **sum** using only one for loop.

```
def mystery(n):
    sum = 0
    for i in range(0, n):
        for j in range(0, n):
            if i == j:
                sum = sum + i * j
    return sum
```

```
def mystery(n):
    sum = 0
    for i in range(n):
        sum = sum + i * i
    return sum
```

Question 6: The Life-Changing Magic of Tidying Up (18 pts)

Lifestyle expert Marie Kondo has created an empire out of helping people throw away any of their possessions that don't spark joy. Let's hop on this bandwagon!

6A. Consonants do not spark joy. (5 points)

Complete the following function that takes a string called `word` as its argument, and returns a new string, which is equivalent to `word`, but with all consonants removed. Y is not a vowel!!! You may assume that the argument `word` consists only of lowercase letters.

```
def no_consonants(word):  
    """...  
    >>> no_consonants('bookkeeper')  
    'ooeee'  
    >>> no_consonants('pry')  
    ''  
    """  
  
    result = ""  
    for letter in word:  
        if letter in "aeiou":  
            result = result + letter  
    return result
```

6B. Double letters do not spark joy. (8 points).

Complete the following function that takes a string called `word` as its argument, and returns a new string, which is equivalent to `word`, but with all “double letters” replaced by just a single letter. More specifically, if the same letter appears **two or more times in a row**, then that letter sequence should be replaced by a single letter. You may assume that the argument `word` consists only of lowercase letters.

```
def no_doubles(word):
    """Replaces all letters that appear consecutively
    with a single letter.

    >>> no_doubles('balloon')
    'balon'
    >>> no_doubles('bookkeeper')
    'bokeper'
    >>> no_doubles('aaaaaaah')
    'ah'
    """

    result = ""
    prev_letter = ""
    for letter in word:
        if letter != prev_letter:
            result = result + letter
            prev_letter = letter
    return result
```

6C. Palindromes absolutely do spark joy. (5 points)

A “palindrome” is a word that is spelled the same backwards and forwards. Complete the following function that takes a string as argument, and returns a `bool` that indicates whether the string is a palindrome (`True` if so, `False` if not). You may assume that the argument `word` consists only of lowercase letters.

```
def is_palindrome(word):  
    """Returns True if the word is a palindrome, and False otherwise.  
  
    >>> is_palindrome('kayak')  
    True  
    >>> is_palindrome('racecar')  
    True  
    >>> is_palindrome('garbage')  
    False  
    """  
    return word == word[::-1]
```

ANOTHER POSSIBLE ANSWER (there are several...)

```
def is_palindrome(word):  
    for i in range(len(word)):  
        if word[i] != word[-i-1]:  
            return False  
    return True
```

Anonymous ID: _____

(This is the end of the exam.)