This is a preview of the published version of the quiz

Started: Feb 28 at 10:43am

**Quiz Instructions**

**Readings**

Please read the following parts of the [Course Notes](https://www.teach.cs.toronto.edu/~csc165h/winter/resources/csc165_notes.pdf).

- Chapter 5, pp. 93-97 (starting at *Back to algorithms*, up to but not including *Loop iterations with changing costs*).

**General instructions**

You can review the general instructions for all prep quizzes at this page. Remember that you can submit multiple times! You might consider printing this quiz out so that you can work on paper first.

---

**Question 1**

Match each function \( g(n) \) on the left with a function \( f(n) \) on the right such that \( g \) is \( \Theta(f) \).

Use each function \( f(n) \) exactly once!

(Due to limitations in Quercus, we are using standard LaTeX notation to represent subscripts and superscripts: “\( n^2 \)” stands for \( n^2 \) and “\( \log_3 n \)” stands for \( \log_3 n \), for example.)

<table>
<thead>
<tr>
<th>( g(n) )</th>
<th>( f(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (n + 3) / n^2 )</td>
<td>[ Choose ]</td>
</tr>
<tr>
<td>( \log_3 (n) + 5 )</td>
<td>[ Choose ]</td>
</tr>
<tr>
<td>( (1/n^2) + 5 )</td>
<td>[ Choose ]</td>
</tr>
<tr>
<td>( n^2 – 10n + 300 )</td>
<td></td>
</tr>
</tbody>
</table>
Question 2

When analyzing the running time of an algorithm, we count the number of "basic operations" performed by that algorithm. How do we define a basic operation?

- One expression in a program (there can be many expressions in a single line of code).
- Any block of code whose running time does not depend on the size of the algorithm's input.
- An arithmetic calculation.
- One line of code.

Question 3

Select every block of code below that constitutes a "basic operation," according to our definition. Assume that each block appears as part of a function with a list parameter named `lst` and that an integer variable `i` has already been defined.

```python
if i % 2 == 0:
    i = i + 1
else:
    i = i * 2
```
Consider the following Python function.

```python
def print_items(lst: list) -> None:
    for item in lst:
        print(item)
```

Let $n$ represent the length of the input list. How many loop iterations occur when we call this function?

- 0
- 1
- $n^2$
- $n$

**Question 5**

1 pts
Consider the following Python function.

```python
def print_items2(lst: list) -> None:
    i = 0
    while i < len(lst):
        print(lst[i])
        i += 2  # Increase i by 2
```

Let $n$ represent the length of the input list. How many loop iterations occur when we call this function? (We're looking for the exact expression here, so please be careful with off-by-one errors.)

- $\frac{n}{2}$
- $\lfloor \frac{n}{2} \rfloor + 1$
- $\lfloor \frac{n}{2} \rfloor$
- $\lceil \frac{n}{2} \rceil$

---

**Question 6**

Consider the following Python function.

```python
def print_items3(lst: List[int]) -> None:
    for item in lst:
        i = 0
        while i < len(lst):
            print(item + lst[i])
            i = i + 2
```

Let $n$ represent the length of the input list. How many loop iterations of the *inner loop* occur when we call this function? Count all iterations of the inner loop, across all iterations of the outer loop.

- $\frac{3}{2}n$
<table>
<thead>
<tr>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n \cdot \left\lfloor \frac{n}{2} \right\rfloor$</td>
</tr>
<tr>
<td>$n + \left\lfloor \frac{n}{2} \right\rfloor$</td>
</tr>
<tr>
<td>$\left\lfloor \frac{n^2}{2} \right\rfloor$</td>
</tr>
</tbody>
</table>