

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Applied Science and Engineering
Division of Engineering Science
PRACTICE TEST

PLEASE HAND IN

ESC 180 H1F

Duration: 50 minutes

Aids Allowed: None

Student Number:

Given Name(s):

Family Name(s):

UTORid:

mail.utoronto.ca email:

*Do **not** turn this page until you have received the signal to start.*
*In the meantime, please read **carefully** every reminder on this page.*

- Fill out your name and student number above—do it now, don't wait!
- Fill out your student number on every odd-numbered page. Failure to do so will result in a 5% penalty.
- This test consists of 5 questions on 7 page (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete.*
- Answer each question directly on the test paper, in the space provided. If you need more space for one of your solutions, use one of the “blank” pages and *indicate clearly the part of your work that should be marked.*
- Comments and docstrings are not required. Comments may help us mark your code. We mark your code and not your comments.
- You may use functions written to answer one question as part of the solution to another question.
- Write your answers in Python. You may **not import** any module other than `math`, `time`, and `random`, **unless otherwise specified**. We will accept code that is correct under any Python 3 standard.
- For full credit, your code should run correctly without crashing. Unless otherwise specified, you will not be penalized if your code runs but is not efficient. Comments are not required.

MARKING GUIDE

Nº 1: ____/10

Nº 2: ____/15

Nº 3: ____/15

Nº 4: ____/15

Nº 5: ____/12

TOTAL: ____/67

Question 1. [10 MARKS]

Write a function that takes in two integers, `a` and `b`, and returns the string `"larger"` if `a` is larger than `b`, `"smaller"` if `a` is smaller than `b`, and `"equal"` if `a` is equal to `b`. For example, `compare(5, 3)` should return `"larger"`, `compare(3, 5)` should return `"smaller"`, and `compare(3, 3)` should return `"equal"`.

```
def compare(a, b):
```

Question 2. [15 MARKS]

Write code that, when run, will print the integers between 180 and 1800, inclusive of 180 and 1800, except that it would not print the number 1000. The numbers must be printed in increasing order: 180, 181, ..., 999, 1001, 1002, ..., 1800. The implementation details are up to you, but the code must **print** what is specified above.

Question 3. [15 MARKS]

Write a function that returns the list of even numbers between the integers **a** and **b**, inclusive of **a** and **b**. For example, `even_numbers(3, 10)` should return `[4, 6, 8, 10]`.

```
def even_numbers(a, b):
```

Question 4. [15 MARKS]

Write a function that takes in a list of strings and returns the number of elements in the list that are not equal to one of "pumpkin", "ghost", or "candy". For example,

`not_halloween(["pumpkin", "ghost", "candy", "pumpkin", "candy"])` should return 0, and

`not_halloween(["pumpkin", "ghost", "candy", "pumpkin", "cat", "candy"])` should return 1.

```
def not_halloween(L):
```

Question 5. [12 MARKS]

Suppose the energy function $E(x_1, x_2, x_3, x_4)$ is defined somewhere in the code (you do **not** need to define it). Assume that x_1, x_2, x_3, x_4 can take on only the values -1 and 1. Write a function that returns sets of values for x_1, x_2, x_3, x_4 that minimize the energy function $E(x_1, x_2, x_3, x_4)$ to within **epsilon**. That is, the function returns a list of lists of values such that $E(x_1, x_2, x_3, x_4)$ is either the energy minimum or is larger than the minimum by at most **epsilon**. For example, the function might return $[[1, 1, 1, 1], [-1, -1, -1, -1]]$.

```
def minimize_energy(epsilon):
```

...

...

...