Duration: **105 minutes**
Aids Allowed: **Reference sheet handed out with the test**

Student Number: |_|_|_|_|_|_|_|_|_|_|_|_|_|

UTORid: _____

Family Name(s): _____

Given Name(s): _____

UofT email: _____@mail.utoronto.ca

---

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

This term test consists of 8 question on 14 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, fill in the identification section above, and write your name on the back of the last page.*

You **must write your student number at the bottom of every odd-numbered page**. **Failure to do so will result in a deduction of 5 points.**

Answer each question directly on the test paper, in the space provided. *indicate clearly the part of your work that should be marked.*

If you need extra space, you can use the last page of the midterm. **Indicate clearly that you have done so if you have**.

When you are asked to write code, *no* error checking is required: you may assume that all user input and argument values are valid, except where explicitly indicated otherwise.

Write up your solutions carefully! Comments and docstrings are *not* required to receive full marks, except where explicitly indicated otherwise. However, they may help us mark your answers, and part marks *might* be given for partial solutions with comments clearly indicating what the missing parts should accomplish.

You may use functions written to answer one question as part of the solution to another question.

You may **not** `import` any modules unless otherwise specified.

Good Luck!

OVER...

Question 1. [5 MARKS]

Write a function that prints `Happy Haloween` if the input month is (exactly, in lower-case) `"oct"` and the input day is (exactly) `31`. Otherwise, the function should print `Not Haloween`.

```
def print_halloween(month, day):
```

Question 2. [10 MARKS]

Write a function that takes in a list of integers and returns the sum of the squares of the even integers in the list. For example, `sum_even_squares([1, 2, 4, 3])` should return `20`, since $2^2 + 4^2 = 20$ (and 2 and 4 are the even integers in the list).

```
def sum_even_squares(L):
```

Question 3. [10 MARKS]

Write a function that returns `True` if the input list of integers is sorted in either increasing or decreasing order. A list with repetitions is not considered sorted.

For example,

```
is_sorted([1, 2, 3, 4, 5]) # True
is_sorted([4, 2, 1])       # True
is_sorted([5, 2, 10])      # False
is_sorted([1, 2, 2, 3])    # False
```

Question 4. [10 MARKS]

Given a name, if the name is one of the following: `"cluett"` or `"stangeby"`, print `trick`. If the name is `"davis"`, print `treat`. Otherwise, print `no candy for you`. Assume that all inputs are in all-lowercase.

```
def print_trick_or_treat_for(name):
```

Question 5. [10 MARKS]

Write a function that takes in a list of strings and a target string, and returns the number of times the target string appears in the list of strings. For example,

`count_occurrences(["candy", "costumes", "midterms", "candy"], "candy")` should return 2, since the string `"candy"` appears twice in the list of strings. You may not use `list`'s `count` method.

```
def count_occurrences(n):
```

Question 6. [10 MARKS]

Write a function that finds the element that occurs the most frequently in a given list of strings. For example, `most_frequent(["candy", "costumes", "midterms", "candy"])` should return `"candy"`, since `"candy"` appears twice in the list of strings, which is more than any other string in the list. Assume there are no "ties". You may not `import` any modules (like in other questions), but may otherwise use any functionality available in Python. You may use helper functions. Hint: one way (but not the only way) to start solving this problem is to count how many times `"candy"` appears and check if it appears more times than any other string in the list, etc.

```
def most_frequent_fave(faves):
```

CONT'D...

Question 7. [10 MARKS]

Write code that would allow a user to call the function `check_next_prime` as follows:

```
check_next_prime(2) # print "Correct"
check_next_prime(3) # print "Correct"
check_next_prime(4) # print "Incorrect, game over"
check_next_prime(5) # print "Game is over"
```

The function prints `Correct` while the user call the function using prime numbers in the order 2, 3, 5, 7, 11, ...

The first time the user enters an unexpected number (e.g., 5), the function prints `Incorrect, game over`. After printing this message, the function only ever prints `"Game is over"`, no matter what the user enters.

Your answer should include the definition of the function, as well as any other code needed to make the code work.

Question 8. [10 MARKS]

A matrix is symmetric if it is symmetric across its major diagonal. For example, the following matrix is symmetric:

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 3 \\ 4 & 3 & 9 \end{bmatrix}$$

Another wait of putting this is that $A_{ij} = A_{ji}$ for all i and j.

Define a matrix is *almost symmetric* if there is a way to swap two of its elements to make it symmetric. For example, the following matrix is almost symmetric:

$$B = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 9 \\ 4 & 3 & 3 \end{bmatrix}$$

That is because we can swap B_{23} and B_{33} to get the symmetric matrix A above.

Recall that a matrix can be represented as a list of lists:

```
B = [[1, 2, 4], [2, 5, 9], [4, 3, 3]]
```

Write a function that returns `True` if the input matrix is *almost symmetric* and `False` otherwise.

```
def is_almost_symmetric(M):
```

Bonus. [5 MARKS]

Recall the Leibnitz formula for computing π: $\pi = 4 \cdot \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$.

Suppose your goal is to display the first 1000 digits of π. This cannot be done with `float`s, but can be done with `int`s. Write code that prints the first 1000 digits of π. The digits should be printed one per line, as follows:

```
3
1
4
...
```

CONT'D...

On this page, please write nothing except your name.

Family Name(s): _____

Given Name(s): _____