Name: ID:

Name: ID:

In-Class Activity # 2

This in-class activity is worth **10% of your final grade**.

You can work in teams of **two**. If you work in team **only one of the team members must submit the activity on Omnivox**.

**Do not forget to write your name and student id at the top of this file**.

With this instruction file, you were given three folder containing each a bug to fix. Each of those folders contains a .py file and a bug report in pdf.

For each bug, there are questions below. You must answer the questions by filling in this file. Some questions ask you to update the .py files to fix the bug.

**What to submit?**

You have to submit **four files** in a **zip file**.

1. **This file** with questions filled in;
2. **truck\_loading.py** fixed version;
3. **html\_formatter.py** fixed version;
4. **counting\_sort.py** fixed version.

Do not change the name of the Python files.

Do not submit each file individually.

**Only the last submission will be graded.**

**Bug 1: truck\_loading.py**

*Answer all the following questions. When providing inputs and outputs from the program or from your debugger, makes sure to* ***use a readable format****.*

1. (10 pts) Give **two inputs** which provided an **incorrect** result and describe the incorrect behaviour of the program.
2. (5 pts) Give **one input** which provided a **correct** result and describe the correct behaviour of the program. If you could not find a correct input, you can provide an extra incorrect input for partial marks.
3. (10 pts) After you started debugging, give the location of one **infection** you found in the program. Give the **line of the infection**, the **name and value of the infected variable** and **explain why** the value is incorrect in one sentence.
4. (10 pts) At what line(s) is the **defect** in the code. **Explain what is wrong** in one or two sentences.
5. (15 pts) After finding the defect, you can fix the code. Submit the fixed code as a .py file.

**Bug 2: html\_formatter.py**

*Answer all the following questions. When providing inputs and outputs from the program or from your debugger, makes sure to* ***use a readable format****.*

1. (10 pts) Give **two inputs** which provided an **incorrect** result and describe the incorrect behaviour of the program.
2. (5 pts) Give **one input** which provided a **correct** result and describe the correct behaviour of the program. If you could not find a correct input, you can provide an extra incorrect input.
3. (10 pts) After you started debugging, give the location of one **infection** you found in the program. Give the **line of the infection**, the **name and value of the infected variable** and **explain why** the value is incorrect in one sentence.
4. (10 pts) At what line(s) is the **defect** in the code. **Explain what is wrong** in one or two sentences.
5. (15 pts) After finding the defect, you can fix the code. Submit the fixed code as a .py file.

**Bug 3: counting\_sort.py**

*Answer all the following questions. When providing inputs and outputs from the program or from your debugger, makes sure to* ***use a readable format****.*

1. (10 pts) Give **two inputs** which provided an **incorrect** result and describe the incorrect behaviour of the program.
2. (5 pts) Give **one input** which provided a **correct** result and describe the correct behaviour of the program. If you could not find a correct input, you can provide an extra incorrect input for partial marks.
3. (10 pts) After you started debugging, give the location of one **infection** you found in the program. Give the **line of the infection**, the **name and value of the infected variable** and **explain why** the value is incorrect in one sentence.
4. (10 pts) At what line(s) is the **defect** in the code. **Explain what is wrong** in one or two sentences.
5. (15 pts) After finding the defect, you can fix the code. Submit the fixed code as a .py file.