

# Team name - Dungeon Overflow

---

Team Member 1: Drew Bosco Email: abosco@eagles.ewu.edu

Team Member 2: Eric Shervey Email: ericbshervey@gmail.com

Team Member 3: Chester Southwood

Contact Drew Bosco for any additional clarification needed. He can also provide the email addresses to the other team members.

---

## OS and GCC

---

### OS Version

Linux Ubuntu 18.04.3 LTS

So if you are not using Ubuntu, you could run into issues. Do not use Windows or Mac with our code.

### GCC

GCC 7.4.0

## Language: Java

---

### Compilation Instructions - Java

1. Unzip folder
2. move inside the folder in the command line
3. compile code by 'javac DefendCode.java'
4. run program by executing 'java DefendCode'

### Screenshot - Java

- [Java Code Output Text File](#)

### Complications we defended against - Java

- **Buffer Overflow:**
  - Protected by using a higher language that handles automatically due to it managing memory and having a garbage collector.
- **Integer Overflow:**

- Protected by having the int values be doubles within the program. The values are checked to make sure that they can be casted to proper integer values if within the range of `INTEGER.MIN_VALUE` and `INTEGER.MAX_VALUE`. The same logic is done for the sum and product, as the max value of a integer multiplied by itself is still within the range of the double.
- **Incorrect data type:**
  - Protected by re prompting the user in a loop for the proper data type via try/catch/finally structure.

## Shortcomings - Java

- User is required to save input file and output file within the source directory, they are unable to specify a outside location due to possible malicious behavior.

---

## Language: C

---

### Compilation Instructions - C

1. Unzip folder
2. move to C Code sub-directory within the command line
3. type the make command to compile the program
4. Execute main program by entering './driver.exe' within the command line.

### Library Dependencies

We have the following C library dependencies that you will need as well:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <unistd.h>
#include <ctype.h>
#include <math.h>
#include <regex.h>
#include <crypt.h>
#include <sys/random.h>
#include <math.h>
```

### Output - C

- [C Output Text File](#)

### Complications we defended against - C

- **Buffer Overflow:**
  - Protected by using safer functions in some cases such as fgetsf and scanf to include null terminating characters within typed strings.
- **Integer Overflow:**
  - Protected by having the int values be doubles within the program. The values are checked to make sure that they can be casted to proper integer values if within the range of INTEGER.MIN\_VALUE and INTEGER.MAX\_VALUE. The same logic is done for the sum and product, as the max value of a integer multiplied by itself is still within the range of the double.
- **Incorrect data type:**
  - Protected by re prompting the user in a loop for the proper data type via a loop using scanf.

## Shortcomings - C

- Names entered by the user must be at least two characters in length.
- User is required to save input file and output file within the source directory, they are unable to specify a outside location due to possible malicious behavior.
- Output file will be created automatically if not already located.
- Decimal values can be entered, but they are promptly converted and tested as integers. No decimal points are kept. They are always rounded down.