COMP2710 project5: Secure Coding

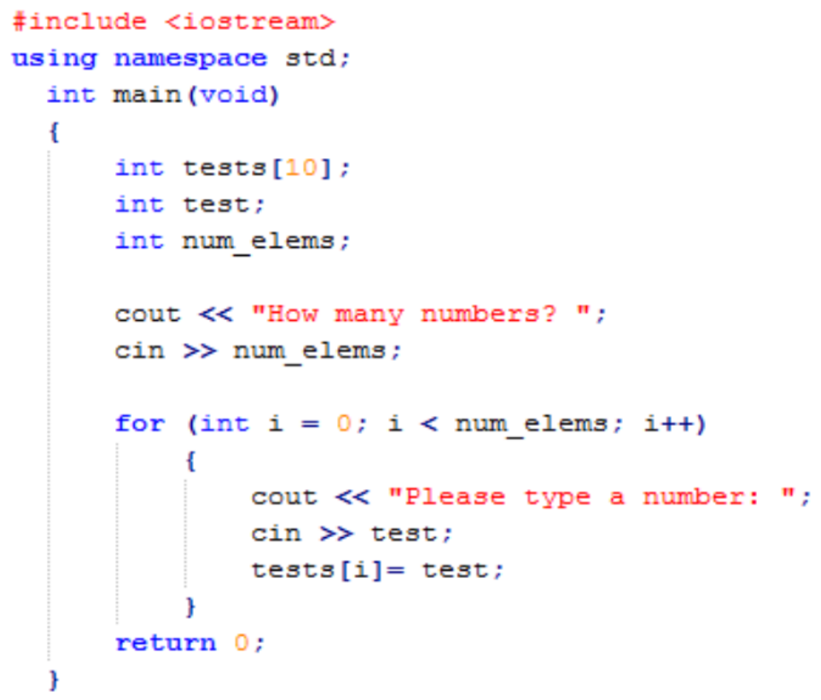
Deadline: 11:59PM Central Time Apr, 21st 2023

Goals:

- To learn how to analyze, identify and resolve buffer overflow
- To compile and run C++ in Linux
- To understand fundamental concepts and principle of secure coding

In this homework assignment, you will test a given example with potential security issues and then answer the following questions.

Sample from Towson University:



```
#include <iostream>
using namespace std;
  int main(void)
  {
      int tests[10];
      int test;
      int num_elems;

      cout << "How many numbers? ";
      cin >> num_elems;

      for (int i = 0; i < num_elems; i++)
          {
              cout << "Please type a number: ";
              cin >> test;
              tests[i]= test;
          }
      return 0;
  }
```

Questions:

1. Describe the buffer overflow problem. (5 points)
2. How could you prevent a buffer overflow from occurring in your program? (5 points)
3. A sample code is presented in a picture format (Above). Please create a secure_coding.cpp and type code with your hands. Compile and Run in AU server(Linux environment). Present your results with a screenshot. (5 points)
4. What happened? (5 points) Why? (5 points) Fix the error and run again. (5 points)
5. Complete the security checklist(below) for this program. (30 points. 5 points each)
6. The V indicates where the potential buffer could occur. How could we prevent this? (10 points)
7. Revise the program to eliminate potential buffer overflow problems. (10 points)
8. Turn in program (marked after completing checklist) (10 points), output(10 points).

Security Checklist

Security Checklist	
Vulnerability: Buffer Overflow **Course:** Software Construction	
Task – Check each line of code	Completed
1. Finding Arrays:	
1.1 Underline each array declaration	
1.2 For each array, underline all subsequent references	
2. Index Variables – *the range i of legal indices for an array of n elements is 0 <= i < n*	
2.1 For each underlined access that uses a variable as an index, write the legal index range next to it.	
2.2 For each index marked in 2.1, underline all occurrences of that variable.	
2.3. Circle any assignments, inputs or operations that may modify these index variables.	▨
2.4. Mark with a V any array that is indexed by a circled index variable.	▨
Highlighted areas indicate a buffer overflow vulnerability.	▨

Note: You will lose **at least 40 points** if there are compilation errors or warning messages when the TA compiles your source code. You will **lose points** if you: do not use the specific program file name, or do not have a comment, or don't have a commend block on program you hand in.

Programming Environment:

Compile and run it using AU server (no matter what kind of editor you use, please make sure your code could run on AU server. The only test bed we accept is the AU server).

Deliverables:

Submit your file named as "Firstname_Lastname_ID.pdf" and code "Firstname_Lastname_ID.cpp" through the Canvas system.

Late Submission Penalty:

- Late submissions will not be accepted and will result in a **ZERO** without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will not accept any late submission caused by internet latency.

Rebuttal period:

You will be given a period of 2 business days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.