CS342/CS343   
Instructor: Professor Izidor Gertner

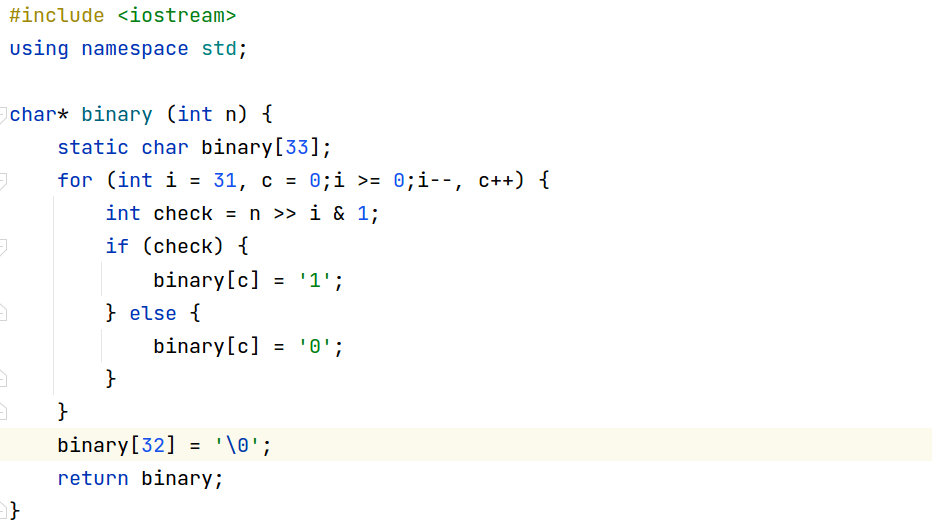
Spring 2022

Homework 2, Azwad Shameem, 2/8/2022

**Objective:**

The objective was to write a program to catch the overflow errors at run time for all possible cases that generate an overflow error.

**Code:**



**Graphical user interface, text, application, chat or text message

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

This program includes functions binary() plus overflow\_check() and also includes the int main(). The int main() image is posted on the right due to lack of space.

Text

Description automatically generated**Results:**

There are five different tests that were tested in int main() and therefore, there are five different outcomes printing out in the results.

**Explanation:**

The code has the functions binary(), overflow\_check() and an int main() which all work together to check for overflow error in the 5 tests. The binary function is quite simple, it bit shifts a decimal into binary and returns the string of the binary result. The overflow\_check() function either prints the decimal and binary values or throws an overflow error and prints it out.

The method the overflow\_check() function does this is by using the 4 cases of overflow. The cases for checking overflow in the overflow\_check() function is the following, positive plus positive equal negative, negative plus negative equal positive, positive minus negative equals negative, negative minus positive equals positive.

In int main(), the functions are tested by 5 equations. The first equation shows a = +2147483647 and b = +1 for output = a+b gives an output = -2147483648, which is wrong because adding two positives cannot equal a negative, therefore it will instead throw an exception and print out overflow error. The second equation shows a = -2147483648 and b = -2 for output = a+b but output = +2147483646, which is wrong because adding two negatives cannot equal a positive, therefore it will instead throw an exception and print out overflow error. The third equation shows a = +2147483647 and

b = -2147483648 for output = a-b but output = -1, which this is wrong because b is negative, and two minuses make a positive. This shows us situation where we are adding two positives, but we get a negative answer which is wrong and therefore, the overflow\_check function throws the exception to print overflow error. The fourth equation shows a = -2147483648 and b = +2147483647 for output = a-b but output here equals 1 which is wrong because we end up with two negative numbers adding which can never be positive, therefore it is an overflow error is thrown and printed. The fifth equation is -1 + +2 and does in fact output +1 which is the right answer and therefore, the decimal and binary outputs are printed instead of an overflow error.

**Conclusion:**

The program to catch overflow errors is very useful because otherwise overflow errors would not be detected. This program allowed me to learn a lot about the edge cases about where an overflow error can occur and opened my mind to a lot of the errors that the computer would just ignore if not dealt with at runtime.