main

Author: Joshua Grizzell
Created: Nov-Dec 2021
Purpose: CIS-7 Fall 2021 Project 2

System Libraries
cstdlib
iostream
ctime
string
iomanip

User Libraries
none

Function Prototypes
string AI(char,char);
bool eval(string,string,char &,char &);
string set();

(main)

(main)

Create random seed
Using srand

Declare variables
string code, guess
char rr, rw
int nGuess

Initialize variables
nGuess = 0
code = set()
rr = rw = 0

(set)   The set function returns a random
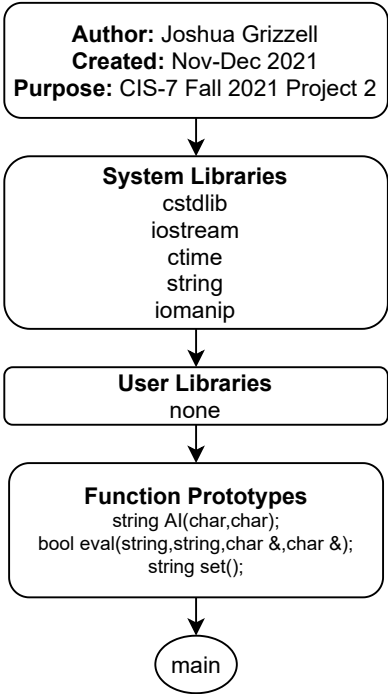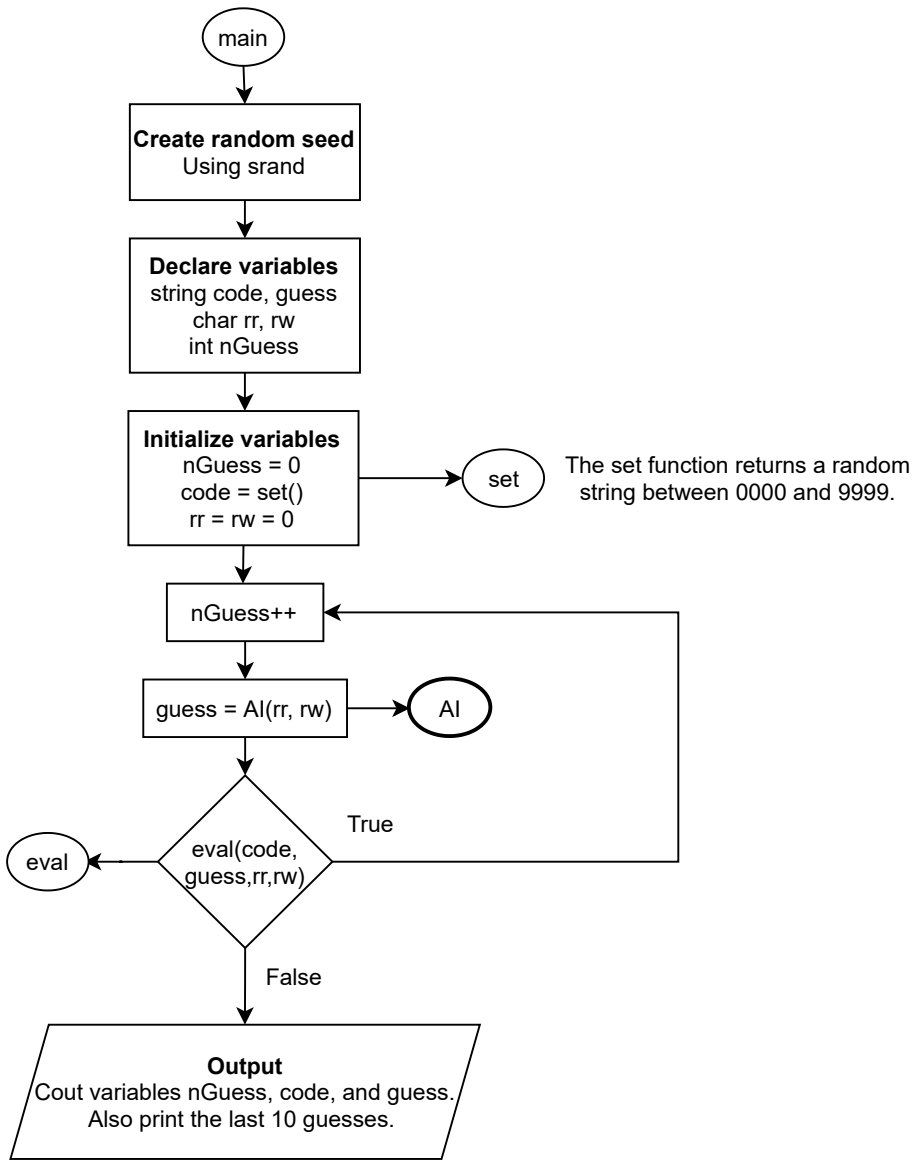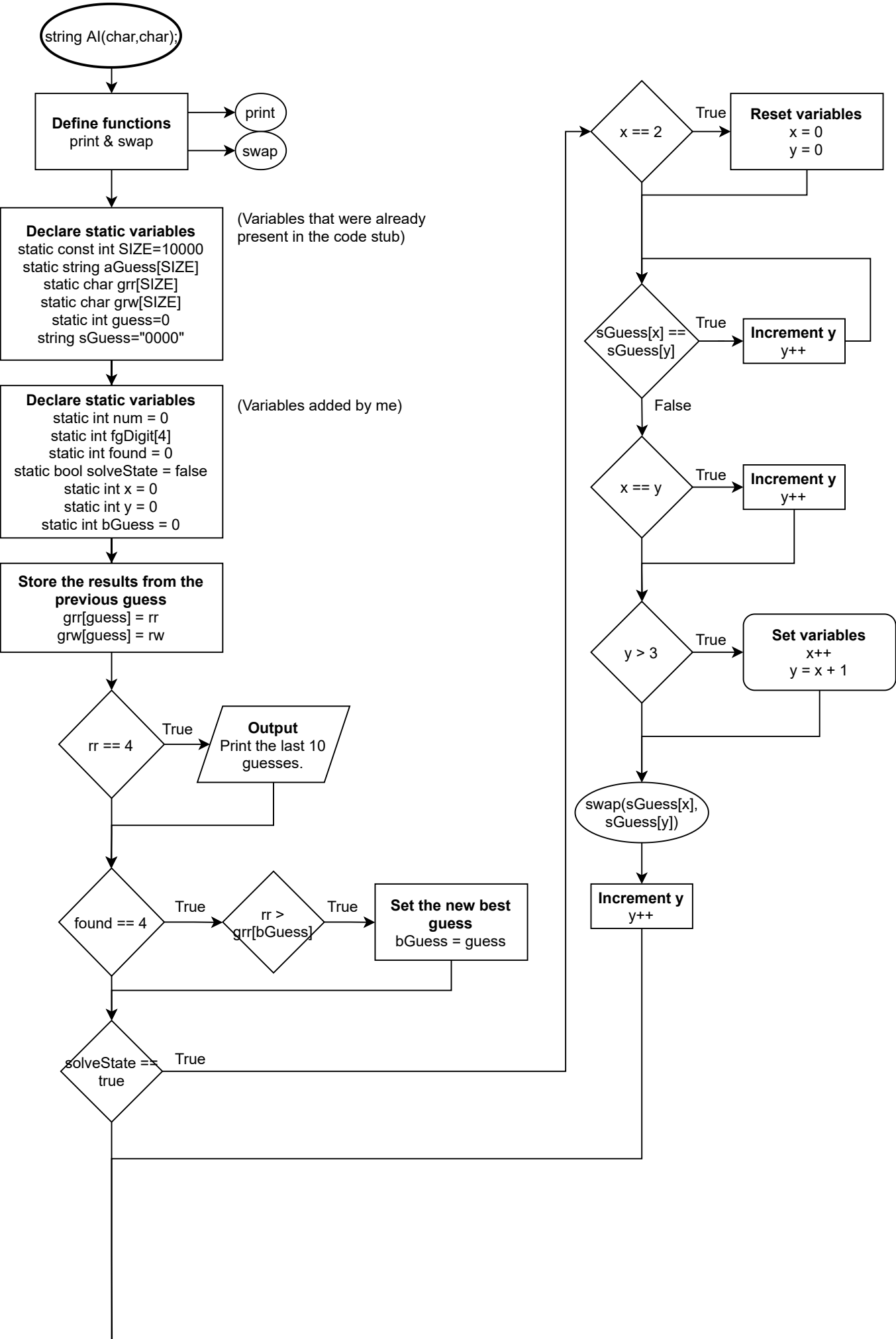string between 0000 and 9999.

nGuess++

guess = AI(rr, rw)   (AI)

The eval function returns false only if all 4
digits are correct. That is, guess == code.

(eval)   eval(code,
guess,rr,rw)   True

False

Output
Cout variables nGuess, code, and guess.
Also print the last 10 guesses.

AI *(part 1)*

```
string AI(char,char);
```

Define functions
print & swap

→ print

→ swap

Declare static variables
static const int SIZE=10000
static string aGuess[SIZE]
static char grr[SIZE]
static char grw[SIZE]
static int guess=0
string sGuess="0000"

(Variables that were already present in the code stub)

Declare static variables
static int num = 0
static int fgDigit[4]
static int found = 0
static bool solveState = false
static int x = 0
static int y = 0
static int bGuess = 0

(Variables added by me)

Store the results from the previous guess
grr[guess] = rr
grw[guess] = rw

rr == 4 —True→ **Output** Print the last 10 guesses.

found == 4 —True→ rr > grr[bGuess] —True→ **Set the new best guess** bGuess = guess

solveState == true —True→

x == 2 —True→ **Reset variables** x = 0 y = 0

sGuess[x] == sGuess[y] —True→ **Increment y** y++

False

x == y —True→ **Increment y** y++

y > 3 —True→ **Set variables** x++ y = x + 1

swap(sGuess[x], sGuess[y])

Increment y y++

AI *(part 2)*

grr[guess] + grw[guess] > 0 && found < 4 — **True** → **Declare & initialize** int queue = grr[guess] + grw[guess] → queue-- — **True** → **Assign a digit to the found digit array** fgDigit[found] = num - 1 → **Increment found** found++

queue-- — **False**

found < 4 — **True** → **Prepare the guess string** sGuess[0] = num+'0' sGuess[1] = num+'0' sGuess[2] = num+'0' sGuess[3] = num+'0'

found < 4 — **False**

found == 4 && solveState == false — **True** → **Prepare the guess string with the found digits** sGuess[0] = fgDigit[0]+'0' sGuess[1] = fgDigit[0]+'0' sGuess[2] = fgDigit[0]+'0' sGuess[3] = fgDigit[0]+'0'

Save the guess string aGuess[++guess] = sGuess

found < 4 → **Increment num** num++

Return the guess string return sGuess