CMSC 201 Fall 2016

GRS Worksheet – Week 3 – Stuck in the Mud

Shall We Play a Game?

This week you and a partner are going to play a simple dice game, and then you will work on translating it into either pseudocode or a flowchart. The game is called "Stuck in the Mud," and the aim is to earn the highest score.

"Stuck in the Mud" Rules:

(Adapted from http://www.activityvillage.co.uk/stuck-in-the-mud)

The rules are simple:

1. Start by rolling all five dice.
2. You may only score points on a roll with no 2s or 5s rolled.
3. Any dice that are 2 or 5 are "stuck in the mud" and can't be re-rolled.
4. Play continues until all five dice are "stuck in the mud."
5. Your final score is the total of all your scored rolls.

Your TA will demonstrate a round or two of the game.

This Little Pig's Turn:

You and your partner will each play a round or two of "Stuck in the Mud," keeping track of your score. As you play, pay attention to the steps you are taking, and start thinking about how you could turn this into an algorithm. (Remember, it needs to be clear, ordered, and complete.)

Planning Your Algorithm:

You and your partner should start brainstorming how you want to express your algorithm – decide together on whether you'll use pseudocode or a flowchart. Write your notes down on paper or the whiteboard near you.

Some important things to consider as you plan:

- What variables will you need? What do you want to call them?
    - How will you store the running total score?
    - What about the dice rolls?
- What sorts of decisions do you need to make during your program?
    - Remember that decisions should evaluate to **`True`** or **`False`**.
    - (We may not have covered decisions in detail yet, but we have seen them in pseudocode and flowcharts.)
- What's your first step? What order do you perform the steps in?
    - How do you know when to stop?

Writing Your Algorithm:

Don’t do this step until you’ve spent some time planning! Remember, don’t “cowboy code” – it will only makes things more difficult later on.

Write down your algorithm on the whiteboard. Remember: be clear, complete, and have your steps in a logical order. For now, don’t worry too much about the exact “code” – something like “Are any of the dice a 2 or a 5?” is fine.

Once you’ve written your algorithm down, you and your partner should “run” it in your heads (or on paper) individually. Follow the flowchart or pseudocode, and double-check that it makes sense. Does your program earn points when a 2 or 5 is rolled? Are you saving your total score somewhere?
If you find something that you think is an error, discuss it with your partner and try to fix the problem before moving on. If you’re stuck, ask another pair for help; if they can’t help either, ask your TA to assist you.

Testing the Algorithms:

Once everyone is done, move to another pair’s whiteboard, and try to use their algorithm. Have one person in your pair be the “player” that rolls the dice and keeps score on paper, and have the other person be the “computer” that gives directions according to the other team’s algorithm. Play one or two games using their algorithm.

Move to a different pair’s whiteboard, and do the same thing, but switch the roles of “player” and “computer” before playing a game or two.

Can you make sense of the other teams’ algorithms? How are they different from yours? Do they still work, even if different? Write your answer below.