Assignment 04 – The Tyranny of Choice

Due: Friday, October 20, 2017 @ 11:59pm

Time to make things a bit more complex.

Unlike the previous problem set, you're allowed to use `if`, `else if`, `else`, `switch`, and compound conditionals in this assignment. Enjoy.

Car AI Challenge

It was my car's fault!

Saw a pretty cool TED talk video about self-driving vehicles. It was thought-provoking. Thought I'd pass that provocation on to you.

After watching the TED video, I dug around a bit and found this site, which provides the basis for this challenge.

Let's assume that the brakes have failed on a (mostly) fully autonomous self-driving car carrying a given number of adult and child passengers. In front of the car, there is a crosswalk with a given number of adult and child pedestrians crossing. To the left of the car is a very solid concrete barrier.

The car needs to make a decision: veer to the left, ploughing into the barrier, killing the car's passengers, or continue going straight, killing the pedestrians in the crosswalk. Pretty grim stuff.

We will evaluate the situation in a very simple manner: assign a base point value for each adult life and each child life and add up the points in the two groups: whichever group has the most points "wins" (i.e. lives). If the two scores are within a given threshold of each other, we will call this a *tough call* and have the car decide which group lives and dies via user input.

MAKE THIS HAPPEN:

1. Prompt the user for, and read in, the number of adults in the car.
2. Prompt the user for, and read in, the number of children in the car.
3. Prompt the user for, and read in, the number of adults in the crosswalk.
4. Prompt the user for, and read in, the number of children in the crosswalk.
5. Ask the user whether the lives of the car's occupants should be valued over the lives of the pedestrians in case of a tough call.
   a. You can assume the user will enter in a 'y' or 'n', but can't be sure of the case of the letter.
6. Output a fatality report, consisting of the following, in order:
   a. The group killed (PEDESTRIANS or OCCUPANTS)
   b. The number of adults killed.
   c. The number of children killed.

NOTES:

- You can assume that adult lives are worth **100 points** and that child lives are worth **145 points**.
- The threshold for a tough call is **10 points**.
- You may use an `enum` (see special topic 3.4 in the text, or do some research) for the two groups (passenger and pedestrian) if you wish. This is not a requirement of the assignment – its use (or lack of use!) will not affect your mark in any way!

Role Playing Game Player Character (RPG PC) Generator Challenge

Great. Strength 3. I can barely draw a breath, let alone a sword.

So I was playing D&D last week, 'cause yeah, some stereotypes are totally justified.

We were just starting a new campaign, so it was time to roll up a new character.

I think you can see where this is going.

You're going to make a tool that takes in 4 numbers (i.e. "dice rolls") and a race, and calculates the **dexterity score** and **dexterity modifier** for the character based on those rolls. Don't worry, if you've never played D&D, I know this will likely sound like a foreign language to you, so there is a quick primer on this comin' up…

SOME BACKGROUND (you can skip this if you know how to play D&D):

So D&D (or Dungeons and Dragons) is an example of a *Role Playing Game*, or *RPG*. To put it very simply, in an RPG, you pretend you're someone else (like a character out of a fantasy novel, or science fiction movie, etc.) and you have some sort of adventure with a number of like-minded people. There are usually some (and by some, I mean way too many) rules involved. And dice - often oddly-shaped ones. And there's usually snacking involved.

Anyway, in D&D, when you want to start an adventure, you have to create your *character*. In the current basic version of D&D, your character can be one of four races:

- human (no surprises here)
- dwarf (think Gimli from the Lord of the Rings movies)
- elf (think Legolas from the same movie)
- halfling (think one of the hobbits from...the same movie)

While you're creating your character, you'll also need to determine his or her *ability scores* - things like how strong she is, how smart, how nimble, etc. To do this in D&D, you roll some dice, add up the numbers, and write that sum down on a sheet for later reference.

You might get to add a small number to your dice sum based on your race as well.

The number you get for each ability score will determine your character's *ability modifier*. If you roll a high sum, you'll get a positive modifier ranging from +1 to +4. If you roll a low sum, you'll get a negative modifier ranging from -1 to -4. You use these modifiers over the course of the game (they are added or subtracted from your rolls), so they're kinda important. Look at the *Ability Scores and Modifiers* table on page 7 of the D&D Player's Basic Rules to see how the ability scores and ability modifiers are related.

For this challenge, we'll deal with the **dexterity** ability - it represents how nimble your character is. A character with high dexterity can dodge things (like arrows or swords being swung at their head), can run across a tightrope like a freaking cat, and are great at hitting things with bows and the like.

That's all the background you need to know for this assignment.

MAKE THIS HAPPEN:

1. Ask the user to enter in their 4 "rolls". Assume they will enter in 4 numbers from 1 to 6. You need to sum up *the highest 3 of those numbers*.
    - *Example:* If the numbers were 3 2 5 5, you would sum up 3, 5, and 5 for a total of 13.
    - You can have these numbers entered one at a time, or all on the same line with tabs or spaces between.
2. Ask the user for which *race* they wish to be - this choice might affect the dexterity sum rolled earlier. To keep things simple, we will ignore subraces (sorry, fans), so here are the only choices you have to worry about, along with the change to the dexterity sum:
    - user enters M or m => **human** => add 1 to the dexterity sum
    - user enters D or d => **dwarf** => no change in dexterity sum
    - user enters E or e => **elf** => add 2 to the dexterity sum
    - user enters H or h => **halfling** => add 2 to the dexterity sum
3. Print to the console the following information in this order (you can format it how you like):
    - Race
    - Dexterity score; if it has been modified by the race, it should have an asterisk next to it.
    - Dexterity modifier (ranging from -3 to +3, and yes, you have to add the + or - sign)
    - If the dexterity score *was* modified by the class, an additional note must be displayed stating the size of the racial increase.
        - The note must be worded exactly like this: **Note: applied racial modifier (+1)**

EXAMPLE OUTPUT:

Let's assume the user has entered the rolls 1,4,1,6 and a race of halfling. Taking the highest 3 numbers (1,4,6), we have a dexterity sum of 11. But because the race was halfling, we add 2 to this sum to get 13. The table on page 7 of the manual says that gives a modifier of +1. So an acceptable output would be:

```
Race: Halfling
Dexterity: 13*
Modifier: +1
Note: applied racial modifier (+2)
```

NOTES:

- You *may* use an `enum` (see special topic 3.4 in the text, or do some research) for the races if you wish. Using - or not using - an `enum` will not affect your mark in any way!
- You **can** assume they will enter in a valid letter (ex. "D" for dwarf)...but you **can't** assume they will enter in a **capital** letter.
- Hint: there is a mathematical way to determine the ability modifier from the ability score.

Submission Instructions

Create a folder named `Asg04_lastname_firstname` and place your BlueJ project (the contents of the `solution` folder) in the folder. Submit your source code to the submit folder (`I:\Labs\CompSci\Submit\1501\00x`). If you are submitting your files via the web interface from off campus, you will need to compress your files into one .zip file.