CSE31: Project #2 – Candy Shop

Overview

You may choose to do this project by yourself or with a partner. The objective of this project is to practice the MIPS coding skills you have learned in the class. You will be implementing a recursive algorithm to solve a puzzle in this project.

Project Details

A candy shop (BobCat Candy) is selling its famous BobCat Bar for ***price*** dollars per bar. In order to promote the product, it will give you an extra bar if you return ***n*** wrappers of the bars.
Your task is to create a **RECURSIVE** algorithm (function) to calculate how many BobCat Bar you can get given the ***price*** of the bar, number of wrappers (***n***) to exchange for a bar, and total ***money*** you have.

For example, if the price of each bar is $2, it takes 2 wrappers to exchange for a bar, and you have $16, you will end up with a total of 15 bars.

Your task:

- Complete the MIPS file called **BobCatCandy.s**
- Fill in the main procedure to read in the three information (***n***, ***price***, and ***money***) from the user.
- Fill in ***maxBars*** function that takes in the information as input arguments. It will return the maximum number of BobCat Bars the user will receive based on the information entered. To solve the problem recursively, you may want to use the helper function (***newBars***), which takes in the ***number of bars*** and ***n*** as arguments and returns a new number of bars based on the input.
- Fill in the main procedure to print out the final statement.
- You **MUST** follow the Register Conventions (see Lecture 11 and Lab #9) when you use any register in this program.
- You will be using a lot of ***syscall*** to print out statements and values. You can refer to http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html to see how to use ***syscall***.
- The output of your program **MUST** follow the **exact format** as the sample output shown below.

Create as many test cases as possible so that your program is free of error.

What to hand in

When you are done with this project, you are ready to submit your work. Make sure you have included the following ***before*** you press Submit:

- Your completed ***BobCatCandy.s***
- A text document containing at least ***5 test cases*** (see examples above)
- In order for your assignment to be graded, you must ***demo*** your code to your instructor **AFTER** submission.

Sample output from BobCatCandy.s:

Test case #1:

```
Welcome to BobCat Candy, home to the famous BobCat Bars!
Please enter the price of a BobCat Bar:
2
Please enter the number of wrappers needed to exchange for a new bar:
2
How, how much do you have?
16
Good! Let me run the number ...
You first buy 8 bars.
Then, you will get another 4 bars.
Then, you will get another 2 bars.
Then, you will get another 1 bars.
With $16, you will receive a maximum of 15 BobCat Bars!
```

Test case #2:

```
Welcome to BobCat Candy, home to the famous BobCat Bars!
Please enter the price of a BobCat Bar:
3
Please enter the number of wrappers needed to exchange for a new bar:
5
How, how much do you have?
30
Good! Let me run the number ...
You first buy 10 bars.
Then, you will get another 2 bars.
With $30, you will receive a maximum of 12 BobCat Bars!
```

Test case #3:

```
Welcome to BobCat Candy, home to the famous BobCat Bars!
Please enter the price of a BobCat Bar:
5
Please enter the number of wrappers needed to exchange for a new bar:
2
How, how much do you have?
4
Good! Let me run the number ...
With $4, you will receive a maximum of 0 BobCat Bars!
```