1811ICT/2807ICT/7001ICT Programming Principles Workshop 9

School of Information and Communication Technology

Griffith University

| Goals | This workshop focusses on everything in the course up to files. |
| --- | --- |
| When | Workshops from Friday 20 May to Thursday 26 May |
| Marks | 3 |
| Due | Pre-workshop questions before the start of the above mentioned workshops  Workshop programming problems by 11:59pm on 29 May |

# Before your workshop class:

* Read all of this document.
* Review the lecture notes sections 1 to 24.
* **Complete the pre-workshop questions (1 mark) posted on the course website and submit the answers for marking**.

# Workshop activities

At any stage, when you are stuck, *ask your tutor*!

## Problem 1

*Problem:* The people of ancient land of Pacific Baza had a simple mathematical system that knew only natural numbers and addition. The genius Bazan scholar, Gringo el Possum, built a computer from wood and various animal parts. Archeologists have recovered ancient scrolls with enough scraps of programs to reconstruct the programming language he named, *Adder*.

The Adder language has only a few simple statements:

| quit | Exit the REPL or terminate a program. |
| --- | --- |
| input *var* | Prompt for and allow the user to enter a value for the variable named *var*. |
| print *val* | Print the value *val*. |
| *var* gets *val* | variable *var* is assigned the value *val*. |
| *var* adds *val* | variable *var* has the value *val* added to it. |

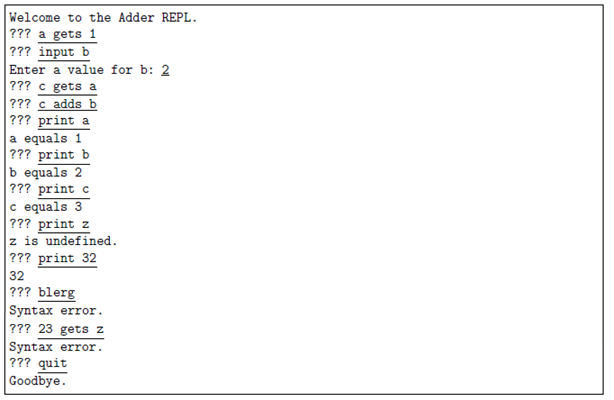
where:

* *var* is always a variable name that contains only letters; and
* *val* can be either:

**–** a variable name that contains only letters; or

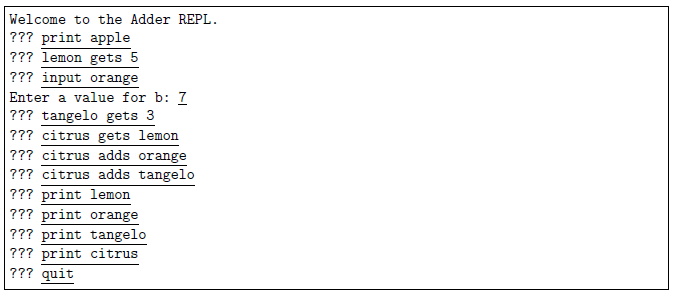
**–** a natural number that contains only digits.

The Adder REPL allows the user to enter commands interactively. For example:



Write the program for the Adder REPL. Hints: Make good use of string methods. Can you divide your program up into smaller pieces by defining functions?

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following scenario:



***Copy your code here***

***def inputfun(var,val):***

***mydic[var] =val***

***return mydic***

***def printfun(var):***

***if var.isdigit():***

***print(var)***

***else:***

***if var in mydic:***

***print(var +" equals "+ str(mydic[var]) )***

***else:***

***print("Undefined")***

***def getsfun(var,val):***

***if val.isdigit():***

***mydic[var] = val***

***return mydic***

***else:***

***if val in mydic:***

***mydic[var] =mydic[val]***

***return mydic***

***else:***

***print("Syntax error")***

***def addsfun(var,val):***

***if val.isdigit():***

***mydic[var] =mydic[var]+val***

***return mydic***

***else:***

***if val in mydic:***

***mydic[var] =int(mydic[var])+ int(mydic[val])***

***return mydic***

***else:***

***print("Syntax error")***

***print("welcome to the adder REPL")***

***comand = input("???")***

***mydic={}***

***while comand!='quit':***

***cm = comand.split()***

***if len(cm) == 2 and cm[0]=='input':***

***val = input("Enter a value of "+ str(cm[1])+":")***

***inputfun(cm[1],int(val))***

***elif len(cm)==2 and cm[0] =='print':***

***printfun(cm[1])***

***elif len(cm)==3 and cm[1]=='gets':***

***getsfun(cm[0],cm[2])***

***elif len(cm)==3 and cm[1]=='adds':***

***addsfun(cm[0],cm[2])***

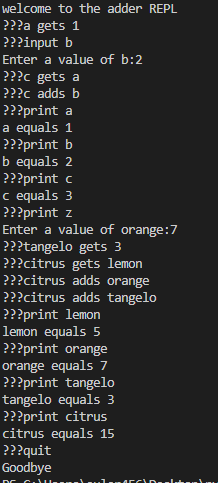
***else:***

***print("syntax error")***

***comand =input("???")***

***print("Goodbye")***

***Insert your screenshot here***



## Problem 2

*Problem:* Write an Adder interpreter, that prompts for and executes an Adder script. For example if the file children.ad contains:

input sons   
input daughters   
children gets sons   
children adds daughters   
print children   
quit

The interpreter would run like this

Script name: children.ad

Enter a value for sons: 3   
Enter a value for daughters: 4   
children equals 7

Hint: This should involve a few small modifications to your REPL from Problem 1.

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following scenario:

* Use the file P2.ad as the source file.

***Copy your code here***

***def inputfun(var,val):***

***mydic[var] =val***

***return mydic***

***def printfun(var):***

***if var.isdigit():***

***print(var)***

***else:***

***if var in mydic:***

***print(var +" equals "+ str(mydic[var]) )***

***else:***

***print("Undefined")***

***def getsfun(var,val):***

***if val.isdigit():***

***mydic[var] = val***

***return mydic***

***else:***

***if val in mydic:***

***mydic[var] =mydic[val]***

***return mydic***

***else:***

***print("Syntax error")***

***def addsfun(var,val):***

***if val.isdigit():***

***mydic[var] =mydic[var]+val***

***return mydic***

***else:***

***if val in mydic:***

***mydic[var] =int(mydic[var])+ int(mydic[val])***

***return mydic***

***else:***

***print("Syntax error")***

***filename=input("Script name:")***

***f1=open(filename)***

***mydic={}***

***for comand in f1:***

***cm = comand.split()***

***if len(cm) == 2 and cm[0]=='input':***

***val = input("Enter a value of "+ str(cm[1])+":")***

***inputfun(cm[1],int(val))***

***elif len(cm)==2 and cm[0] =='print':***

***printfun(cm[1])***

***elif len(cm)==3 and cm[1]=='gets':***

***getsfun(cm[0],cm[2])***

***elif len(cm)==3 and cm[1]=='adds':***

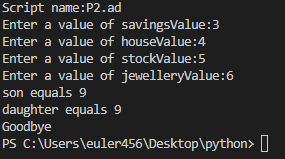
***addsfun(cm[0],cm[2])***

***elif comand=='quit':***

***break***

***print("Goodbye")***

***Insert your screenshot here***



## Problem 3

*Problem:* A road map defines locations as map references like B3, where B is the *x*-coordinate value and 3 is the *y*-coordinate.



The grid lines are 0.5 km apart.

Write a program that allows the user to enter a trip as a sequence of any number of map references on one line, and reports the total length of the trip, assuming they can travel in straight lines. For example:

Enter trip map references: C2 B5 Y25

Total distance = 16.8 km

For badly formatted map references, your program should exit, reporting the first bad map reference.

Enter trip map references: E6 E4 D7 d43 F5

Bad reference: d43

Hints: You need to *split* the input line into separate references; each reference starts with one character which must be an upper case letter, and the rest must be only digits; and Pythagoras will help. The function exit() can abort the program if you detect an error in the input.

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

* A1 B2 C3 D20 S15 W25 Z26
* D2 F23 Ja E23 Z2

***Copy your code here***

***Insert your screenshot here***

# Submission and marking

The pre-workshop can be accessed and submitted online using the provided link in the course website. Students get 1 mark if they get >50% in pre-workshop questions, or 0.5 mark if they get 0%-50% in pre-workshop questions, or 0 marks without any attempt.

For workshop tasks, please submit this document with copied codes and inserted screenshots using the provided submission link in the course website. Students get 2 marks if they complete two or more problems correctly, or 1 mark if they complete one problem correctly, or 0 marks without any attempt.