Computer Organisation COMP2008

Lab Sheet 5 (starts session week 6, due in week 7)

Student Name and Number	
Date, Grade and Tutor signature, **max mark 3**	

Keep this cover sheet marked and signed by the tutor.

1. Preparation [Total max. mark: 1]

The main goal of today's lab is to become familiar with manipulating arrays using pointers and indices, and to reinforce the high level language idioms introduces in the lectures. References: the textbook Ed2, paragraphs 3.5, 3.11 or Ed3: 2.6, 2.15, Ed4: 2.7, 2.14, Ed5: 2.7, 2.14, Ed6: 2.7, 2.14, and the lecture notes.

1. Describe below in writing (or in tabular format) INDICES and POINTERS methods of ARRAY traversal by comparing their significant aspects. [**0.5 mark**].
   Hint: Refer to lecture 3 relevant notes and recall the construction of loop structure. It's also helpful to work on this question together with Workshop Tasks -> Item 2 (below in this lab sheet) to explore the technique for array traversal.

2. Illustrate your explanation on array traversal using INDICES and POINTERS methods with a hand drawing and a few lines of code [**0.5 mark**].

2. Workshop Tasks [Total max. mark: 2]

1. Create a subfolder for lab 5 and copy assembly language files *positivesum.s* and *search.s*.

2. Invoke the PCSpim simulator. Run the programs *positivesum.s,* and then *search.s*. Experiment with these programs to understand how they work. Analyse **different ways** in which they traverse the array.
   Recall the Preparation questions: Compare the methods of array traversal used in two example programs.

3. **Task I (1 mark):** write your own MIPS program that allocates space for two integer arrays (P and Q) of size 9 each. You can **use pointers or indices** for traversing arrays. The program should do the following:

   a. Read from the keyboard for the value of N (a number between 1 and 9, which is used as the actual working size of your array).

   b. Read from the keyboard for the first N values (i.e. N elements) of the array P. Note that we want both arrays P and Q to have identical elements (see the example below for additional explanation).

   c. When all the readings in **step b** are completed, display the N values (i.e. N elements) of both arrays P and Q on the console (use any appropriate output format and additional annotation you like).

   Hint: use a loop.

   Example: let's say that the digit read from the keyboard was 3. This means that N=3; them our program will initialise two arrays with elements P[0], P[1], P[2] and Q[0], Q[1], Q[2].
   Elements of the arrays are to be read from the keyboard, let's assume that they were: 25, 11 and 2. The arrays are to be initialised as following:
   Q[0]=25 Q[1]=11 Q[2]= 2
   P[0]=25 P[1]=11 P[2]= 2

   The best approach is to write a small program implementing only task 3a as described above, and once you are happy that it works correctly, add code to your program to perform task 3b, and so on.

   Demonstration: Demonstrate to the tutor your running program, and be prepared to explain the code.

4. **Task II (1 mark):** Modify your program in Task I (save it with different name) to build and print on the console the arrays P and Q as follows:

   a. Check that "N" entered from the keyboard is within the given range (1 to 9), if it is out of range, print prompt message on the console requesting the next input until getting a valid number.

   b. P – as in Task I above;
      Q – elements are calculated as the accumulation of the array P, e.g. Q[i] = P[0] + P[1] + ... + P[i].

   c. When all the readings in **step b** are completed, display the N values (i.e. N elements) of both arrays P and Q on the console (use any appropriate output format and additional annotation you like).

   Example: let's say that numbers read from the keyboard are the same as in the previous example: N=3, elements are 25, 11, 2. This means that our program has to initialise two arrays: P[0], P[1], P[2] and Q[0], Q[1], Q[2]. Array P will be the same as before, array Q will be build as follows:
   Q[0]=P[0]=25 Q[1]=P[0]+P[1]=25+11=36 Q[2]= P[0]+P[1]+P[2]=25+11+2=38

   Demonstration: Demonstrate to the tutor your running program, and be prepared to explain the code.

3. Assessment notice

When you ready, present to the tutor a printed copy of your program source code, with your name and student number included in the comments (#…), and typed or neatly written answers to all questions listed in the lab sheet. Your tutor may decide to keep the source code printout, but you should keep marked and signed cover sheet.

Warning: Any source code duplicated amongst students will result in a zero mark, and possible further action according to the WSU policy on plagiarism.