MIPS Programming Lab Assignment 2

Q1. Write a MIPS assembly for the following C codes :

a.

```
int a, b, result
int main(){
   a = 0x12345;
   b = 7;
   result = a + b;
   return 1;
}
```

b.

```
 int i;
int b[500];
int a[500];
for(i=0; i < 500; i++) {
 b[i] = a[a[i]];
}
```

In each case check whether you have generated the correct assembly by "cross-compiling" the c – code for MIPS architecture.

[Hint : You will need a gcc tool chain for cross compilation - in ubuntu you can get it using :

sudo apt-get install --install-recommends gcc-mips-linux-gnu cpp-mips-linux-gnu

Once the tool chain is installed search the internet to learn how to cross-compile a code for or a target processor (MIPS in this case)]

Q2. Download a copy of the program *buggy.s* provided in the helpful resources section

a. Inspect the code and try to understand what each register is being used for.

b. Run the code and see what it does.

c. Using MARS, single-step through the code. Watch the registers and see whether they're changing the way you expect. Note carefully each statement that executes without an exception.
d. When an exception does occur, clear everything and reload the code. Step up to the instruction just before the exception and think about what is supposed to happen next.

There are at least two errors in this code. What are they? Write a short description of the errors and place it in a text file, bugs.txt.

Q3. Write an MIPS assembly program that checks if a 32-bit number is a palindrome. Assume that the input is available from the user through the console.

The program should return " Yes a palindrome" if the input is a palindrome else should return " No, not a palindrome"

Q4. Write a MIPS program that given a number *N* and *N* integers can print the integers in a sorted order using *Bubble Sort*. Bubble Sort algorithm involves swapping of two numbers. Write a procedure for swapping two numbers separately and use it in the sort function.

Q5. *Binary Search* is an algorithm to search a value in a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty. (NISTDefinition)Suppose that a sorted array of positive integers is stored in the memory. Write a MIPS program that accepts a positive integer from a user and returns the leftmost index of the element in the array if the element is present in the array. Otherwise, the program returns−1

Q6. Write an ARM assembly program to find out if a number is prime using a recursive algorithm