Introduction
The objective of this lab is to write a short program in MIPS assembly code and simulate it using SPIM. This will familiarize you with the MIPS instruction set architecture (ISA).

Design Requirements
Write a program in MIPS assembly language that implements the bubble sort algorithm to sort a variable-sized array of signed 32-bit integers (words) that are read from the console. A “special value” of 9000 will be used to signify the end of the input sequence. This value is not to be considered part of the input data set. However, any value > 9000 that is entered prior to 9000 is still valid. Zero and negative values are also valid. Empty input sets are also valid.

Use the following algorithm, shown in Java-like syntax:

```plaintext
n=0;
read in;
while in != 9000 {
  vals[n]=in;
  n++;
  read in;
}

for (i=0;i<n-1;i++) {
  for (j=0;j<n-1;j++) {
    if (vals[j] > vals[j+1]) {
      // swap
      temp=vals[j];
      vals[j]=vals[j+1];
      vals[j+1]=temp;
    }
  }
}

for (i=0;i<n;i++) {
  print vals[i]
}
```

System call 5 ($v0=5) reads an integer from the console (returns to $v0), while system call 1 ($v0=1) prints an integer to the console (input in $a0).

Use the following line to set up memory to hold the input:

```plaintext
.data
vals: .space 4000
```

What to Submit
Submit your code via CSE Dropbox (via https://dropbox.cse.sc.edu)