

University of California, Santa Barbara
Dept. of Electrical and Computer Engineering
ECE 15B – Computer Organization
Homework #4
REVISED May 26, 2006

1. Convert the following numbers to single precision (32 bit) IEEE 754 floating point numbers. Put your answer in hexadecimal notation
 - a. 1536
 - b. .015625
 - c. 245.125

2. Convert the following hexadecimal numbers to their MIPS instruction:
 - a. 0x3C019000
 - b. 0x10040004
 - c. 0x8c240180
 - d. 0x0C000000

3. Convert the MIPS instructions into a hexadecimal number
 - a. ori \$2, \$0, 10
 - b. sw \$s3, 284(\$t0)
 - c. j 0x12345678
 - d. srl \$at, \$sp, 9

4. Convert the following C function to MIPS assembly code (using only real MIPS instructions). Draw or describe the order of the variables on the stack.

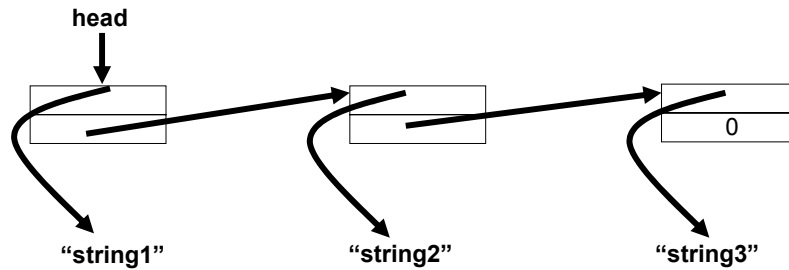
```
void swap(int *px, int *py){
    int temp;
    temp = *px;
    *px = *py;
    *py = temp;
}
```

5. Convert the following C function to MIPS assembly code (using only real MIPS instructions). Draw or describe the order of the variables on the stack.

```
int strlen(char *s)
{
    char *p = s;
    while(*p != 0)
        p++;
    return p-s;
}
```

6. How many bytes do the following structures require in MIPS?
 - a. struct point { int x; int y; }
 - b. struct rectangle { struct point pt1; struct point pt2 ;}
 - c. struct tree { int value; struct tree *left; struct tree *right; }
 - d. struct stringList{ char * value; struct stringList * next;}

7. A stringList is a list data structure where the values are a char * (string). It is defined exactly the same as the structure in problem 6.d. Furthermore, define “head” as the pointer to the first element of the string list. A sample stringList looks like this:



You must translate that following memory layout into a stringList similar to the one above. The variable “head” is 0x00001000. The strings are coded as ASCII characters, e.g. ‘A’ = 0x41. You can find hex to ASCII tables easily by searching the web.

Address	Value	Address	Value
0xFFFFFFFF	.		0x00000010
	.		
	.		
	0x00008000	0x00030000	0x00008000
	0x52455455		.
			.
			.
0x4024FFA4	0x504D4F43		0x4024FFA4
	.	0x00008000	0x00423531
	.		.
	.		.
	0x00030000		
	0x4E4F4954		0x00030000
	0x415A494E	0x00001000	0x00238480
0x0A00C84C	0x4147524F		.
	.		.
	.		.
	0x0A00C84C		0x00238480
0x00238480	0x00454345	0x00000100	0x00008000
	.		.
	.		.
	.		
	0x00000000		0x00041000
0x00041000	0x0A00C84C	0x00000010	0x4024FFA4
	.		.
	.		.
	.	0x00000000	

