Question 1 (10 pts): Variable Names

Indicate if each is a valid (Yes), or not valid (No) variable name

a. _YES_  For
b. _NO_  !notTrue
c. _YES_  dayOfWeek
d. _NO_  for
e. _YES_  mylist1
f. _NO_  seventy_Miles/Hour
g. _NO_  70MilesAnHour
h. _YES_  __height
i. _YES_  object_height
j. _NO_  week#
Question 2 (10 pts): Memory Allocation (Reminder assume an ATMega128 is the target hardware unless otherwise specified)

a. For each declaration indicate how many bytes will be allocated in memory (5 pts)

   i) 90 unsigned int idNumber[45];
   
   ii) 1 char first_letter;

   iii) 1000 signed short altitude[500];

   iv) 120 unsigned long timeCode[30];

   v) 4 double distance;

b. For each indicate the value of my_length after executing each fragment of C code. Function strlen is a standard C library function. Specify N/A if the value of my_length cannot be determined. (5 pts)

   i) my_length is 15.

   char msg[] = “Atmel ATmega128”;
   int my_length = 0xFFFF;
   my_length = strlen(msg);

   ii) my_length is 12.

   char msg[20] = “16 MHz Clock”;
   int my_length = 0xFFFF;
   my_length = strlen(msg);

   iii) my_length is 5.

   char msg[] = {‘A’,’T’,’M’,’E’,’L’,’\0’,’1’,’2’,’8’};
   int my_length = 0xFFFF;
   my_length = strlen(msg);

   iv) my_length is 11.

   char msg[] = “Atmel ATmega128”;
   int my_length = 0xFFFF;
   my_length = strlen(msg + 4);
v) my_length is __N/A__.

```c
char msg[7] = {'C','P','R','E','2','8','8'};
int my_length = 0xFFFF;

my_length = strlen(msg);
```
Question 3 (10 pts): Memory Map

Fill in the given memory map after the C fragment below has been executed. Note: The ATMega128 uses Little Endian ordering for memory (i.e. lower significant bytes of an element are stored at lower addresses) (8 pts)

```c
int class[2] = {0x0285, 0x0281};
long time_stamp[1] = {0x44332211};
char speed[7] = "016MHz";

class[5] = 0x4847;
time_stamp[1] = 0x47303633;

printf("%s", speed);
```

<table>
<thead>
<tr>
<th>Memory Location</th>
<th>0xFF00</th>
<th>0xFF01</th>
<th>0xFF02</th>
<th>0xFF03</th>
<th>0xFF04</th>
<th>0xFF05</th>
<th>0xFF06</th>
<th>0xFF07</th>
<th>0xFF08</th>
<th>0xFF09</th>
<th>0xFF0A</th>
<th>0xFF0B</th>
<th>0xFF0C</th>
<th>0xFF0D</th>
<th>0xFF0E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0x85</td>
<td>0x02</td>
<td>0x81</td>
<td>0x02</td>
<td>0x11</td>
<td>0x33</td>
<td>0x44</td>
<td>0x33</td>
<td>0x36</td>
<td>0x30</td>
<td>0x47</td>
<td>0x48</td>
<td>0x7A</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td>class</td>
<td>time_stamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) What message will `printf` print for part a (2 pts)

360GHz
Name: Lab Section:

**Question 4 (10 pts): C-string formatting**

Given:

```c
char msg[100];
char str1[] = “ATMega”;
char str2[] = “CPRE 288”;
int num = 128
int neg_one = -1;
char ch1 = 77;
char ch2 = 97;
```

Give the C-string contained in `msg` after each `sprintf` (2 pts each)

**Note 1:** Treat each part independently

**Note 2:** You may need to look up on your own more details on `printf` and `sprintf`

**a)** `sprintf(msg, “In lab you program an %s%d.”, str1, num);`

In lab you program an ATMega128

**b)** `sprintf(msg, “%s has %d characters.”, str2, 8);`

CPRE 288 has 8 characters

**c)** `sprintf(msg, “%d in hex is %X.”, neg_one, neg_one);`

-1 in hex is FFFF.

**d)** `sprintf(msg, “%s %c%c%c, best video game!”, str1+2, ch1, ch2, ch1+1);`

Mega MaN, best video game!

**e)** `sprintf(msg, “CPRE 288 makes students %c%c%c”, 58, 45, 41);`

CPRE 288 makes students :-)
Question 5 (10 pts)
Follow the “Quick Linux Tutorial” given for HW2 to see how to write a C program in a text editor, and to compile and run your program from the Linux operating system command line.

Your program should print the following:

Hello, my name is FIRST LAST!

Where FIRST and LAST should be your first and last name. The graders will check this against the name on your homework.

Provide the following screen shots

a) A screen shot of your completed source code within a Linux text editor (5pts)

```c
#include <stdio.h>
void main() {
    // Note: FIRST and LAST is the student's name
    printf("Hello, my name is FIRST LAST!\n");
}
```

b) A screen shot that shows i) the gcc command to compile your program, ii) the command to run your program, and iii) the output of your program. (5pts)

```
[imciner2@linux-4.ece.iastate.edu ~]$ gcc -o question5 question5.c
[imciner2@linux-4.ece.iastate.edu ~]$ ./question5
Hello, my name is FIRST LAST!
[imciner2@linux-4.ece.iastate.edu ~]$ 
```
Question 6: Struct and Union (10 pts)

What is the size of the following data structures in bytes, assuming we are using the Atmega128?

a) Size in bytes: 10 (2 pts)

```c
struct student_record {
    signed int age;
    unsigned long isuID;
    double gpa;
};
```

b) Size in bytes: 4 (2 pts)

```c
union val {
    int size
    char color;
    long time_stamp;
    char first_last_initial[2];
};
```

c) Size in bytes: 8 (3 pts)

```c
struct compound {
    char *name;
    short *age;
    union {
        char *address;
        long distance;
        float *temperature;
    } u;
};
```

d) Size in bytes: 19 (3 pts)

```c
struct more_compound {
    char *name;
    short age;
    long pay[2];
    float *cost;

    struct {
        char red;
        char blue;
        char green;
    } c;
};
```
Name: Lab Section:

    } pixel;

    union {
        char *text_data;
        int  *numeric_data;
        long *l_numeric_data;
    } data;

}