CprE 288 Sprint 2014 – Homework 3
Due Fri. Feb. 7

Notes:
- Homework answers must be typed using a word editor.
- Homework is individual work. Adhere to the University’s policy relating to the integrity of scholarship. See http://catalog.iastate.edu/academiclife/#regulationstext, “Academic Dishonesty”.
- Late homework is accepted within three days from the due date. *Late penalty is 10% per day (for each 24 hours following your class).*

Note: Unless otherwise specified, all problems assume the ATMega128 is being used

Question 1: Function and Pointer (10 pts)

The following function is intended to calculate the sum and product of two integers in a single function call:

```c
// calculate the sum and product of two integers
void sum_and_product (int sum, int product, int x, int y)
{
    sum = x + y;
    product = x * y;
}

// example of calling sum_and_product()
void call_example()
{
    int sum, product;
    int a = 10, b = 20;

    ... // some code

    sum_and_product (sum, product, a, b);

    ... // some code, e.g. print out sum and product
}
```

a. [5 pts] The function does not work as intended. Explain Why.
Because of the parameter of the function trying to pass the parameter by value. The function receives copies of these values and accesses them by the identifiers sum and product. It did not changed the passed value.

b. [5 pts] Revise the code, including both sum_and_product and call_example functions. Test your code, and then cut and paste it in the space below. Highlight your changes in bold type.

```c
// calculate the sum and product of two integers
void sum_and_product (int *sum, int *product, int x, int y) {
    *sum = x + y ;
    *product = x * y ;
}

// example of calling sum_and_product()
void call_example() {
    int sum, product;
    int a = 10, b = 20;

    ... // some code

    sum_and_product (&sum, &product, a, b);

    ... // some code, e.g. print out sum and product
}
```

Question 2: Struct and Union (10 pts)
What is the size of the following data structures in bytes, assuming we’re using the Atmel compiler?

a) 2 pts.
Answer: 5 bytes
```c
struct point3D {
    short x;
    unsigned short y;
    char z;
};
```

b) 2 pts.
Answer: 8 bytes
```c
union val {
    char cval;
    char str[8];
};
```
int ival;
float fval;
double dval;
};

c) 3 pts.
Answer: 8 bytes
struct compound {
  char *mystring;
  int utype;
  union {
    char c;
    int i;
    float f;
  } u;
};

d) 3 pts.
Answer: 18 bytes
struct more_compound {
  char *name;
  int *age;
  long pay;
  long height;

  union {
    char short_id;
    int normal_id;
    long long_id;
  } id;

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

Question 3: Pointers (20 pts)

Complete the table below (i.e. fill in the memory map) to show the state of memory after the following C fragment has been executed. Assume the ATmega128 platform, and all the variables are in the stack (and thus the storage is allocated from high address to low address).

Be very careful with this question.
```c
typedef struct pixel
{
    unsigned char red;
    unsigned char green;
    unsigned char blue;
    unsigned char psize;
} pixel_t;

char num_array[3] = {5, 10, 15};
unsigned char* ptr_array;
unsigned char y = 20;
unsigned char x = 30;
pixel_t my_pixel = {25, 15, 55, 40};
pixel_t *pixel_ptr = &my_pixel;

ptr_array = num_array - 4;
x = *ptr_array;

ptr_array++;
y = *ptr_array;

ptr_array += 3;
*ptr_array = 20;

pixel_ptr++;
pixel_ptr++;
pixel_ptr++;
pixel_ptr->green = 60;
```

<table>
<thead>
<tr>
<th>Address</th>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x010C</td>
<td>num_array[2]</td>
<td>15</td>
</tr>
<tr>
<td>0x010B</td>
<td>num_array[1]</td>
<td>60</td>
</tr>
<tr>
<td>0x010A</td>
<td>num_array[0]</td>
<td>20</td>
</tr>
<tr>
<td>0x0109</td>
<td>ptr_array</td>
<td>0x010A</td>
</tr>
<tr>
<td>0x0108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0107</td>
<td>y</td>
<td>20</td>
</tr>
<tr>
<td>0x0106</td>
<td>x</td>
<td>30</td>
</tr>
<tr>
<td>0x0105</td>
<td>my_pixel.psize</td>
<td>40</td>
</tr>
<tr>
<td>0x0104</td>
<td>my_pixel.blue</td>
<td>55</td>
</tr>
<tr>
<td>0x0103</td>
<td>my_pixel.green</td>
<td>15</td>
</tr>
<tr>
<td>0x0102</td>
<td>my_pixel.red</td>
<td>25</td>
</tr>
<tr>
<td>0x0101</td>
<td>pixel_ptr</td>
<td></td>
</tr>
<tr>
<td>0x0100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 4: String and pointers (10 pts)

a. [5] The following function counts the number of characters in a C string excluding the NULL character, i.e. the length of the string. Complete the function using array access, e.g. s[i], but no pointer access.

```c
int strlen(char s[])
{
    int i, sum;
    sum = 0;
    while(s[i]!="\0"){
        sum++;
        i++;
    }
}
```

b. [5] Rewrite the above function, this time using pointer access, e.g. *s, but no array access.

Note: In a function, parameter declarations “char str[]” and “char *str” are equivalent. You may declare a parameter as an array and use it as a pointer, or vice versa.

```c
int strlen(char *s)
{
    int sum;
    sum=0;
    while(*s!='\0'){
        sum++;
    }
    return sum;
}
```
Question 5: Switch Statement (10 pts)

Complete the following function that counts the prime numbers under 10, i.e. 2, 3, 5, 7, in an array of integers. **You have to use a switch statement.**

```c
// Count prime numbers under 10. X is the array, size is the array size. Return the count of those prime numbers.
int count_prime(int X[], int size)
{
    int i;
    int count=0;
    for(i=0;i<size;i++)
    {
        int number = X[i];
        switch(number){
            case 2:
                count++;
                break;
            case 3:
                count++;
                break;
            case 5:
                count++;
                break;
            case 7:
                count++;
                break;
        }
    }
    return count;
}
```