CS-2110 Quiz 5A

Alan Chiang

TOTAL POINTS

100 / 100

QUESTION 1

1 Pointer Tracing 25 / 25

+ 0 Graded
+ 5 b = 10
+ 5 a = 12
+ 5 pa = &b or pa = pb
+ 5 b = 33
+ 5 b = 12

QUESTION 2

2 Pointer Arithmetic 20 / 20

+ 0 Graded
+ 5 1a: 0x06000005
+ 5 1v: 0x05
+ 5 2a: 0x06000007
+ 5 2v: 0x0708

QUESTION 3

3 Struct Typedef 10 / 10

+ 0 Graded
+ 5 struct correctly declared - ex: typedef struct { ... } FOO; also correct - ex: typedef struct anything { ... } FOO;
+ 5 3 Ints declared inside struct - bar, baz, and qux

QUESTION 4

4 Macro 10 / 10

+ 0 Graded
+ 2.5 Parentheses around parameters
+ 2.5 Left Shift first parameter by 16
+ 2.5 Use bitwise OR "|" or arithmetic ADD "+" to combine parameters
+ 2.5 Correct Macro Format - ex: #define PACK(a,b) (...)
+ 0 Click here to replace this description.

QUESTION 5

5 Swap 25 / 25

+ 0 Graded
+ 5 swap called correctly - ex: swap(&a, &b)
+ 5 swap function takes in 2 int pointers - ex: void swap(int *a, int *b)
+ 5 value of "a" or "b" stored in temporary variable
+ 10 Works

QUESTION 6

6 Short Answer (Static Keyword) 10 / 10

+ 0 Graded
+ 4 static variable in a function - variable is initialized once, shares its value every time its called
+ 3 static on function - function is only accessible from the file (or compilation unit)
+ 3 static on global variable - variable is only accessible from the file (or compilation unit)

CS-2110 Fall 2016 Quiz 05a

Name: Alan Chiang Section: 3:00

Pointer Tracing (___/25)

For each line in the following table, show what changes after it is executed. **You don't need to write each value on each line, only the values that changed on that line.** The first few have been done for you.

Statement	a	b	pa	pb	ppa
int a = 10;	10				
int b = 25;		25			
int *pa = &a;			&a		
int *pb = &b;				&b	
int **ppa = &pa;					&pa
b = *pa;		10			
*pa = 12;	12				
*ppa = pb;			&b		
**ppa = 33;		33			
*pb = a		12			

alias

pa = pb = &b

**pa = 33

Pointer Arithmetic (___/20)

Use the memory view below to answer the following 2 pointer arithmetic questions. Note that the value at memory address `0x06000000` is `0x00`, the value at `0x06000001` is `0x01`, the value at `0x06000002` is `0x02`, etc. Also, assume that this memory is **big endian**, meaning the value of an `int` located at `0x06000000` is `0x00010203`.

```
0x06000000  0x00  0x01  0x02  0x03
0x06000004  0x04  0x05  0x06  0x07
0x06000008  0x08  0x09  0x0A  0x0B
```

```
short* aaa = (short*) 0x06000001 + 4
int*   bbb = (int*)   (aaa + 5) + 10 - 4
```

0x0600000B

For each of the questions below, calculate the memory address of the pointer, and the value it's pointing at. Assume that a **char = 1 byte**, a **short = 2 bytes**, and an **int = 4 bytes**. Write the address as an 8 digit hexadecimal number, and the value with the fewest number of digits required to represent the data type. For example: a char (1 byte) can be represented with 2 hexadecimal digits.

1) `(char*) (aaa + 2)` address: 0x06000005 value: 0x05

2) `(short*) (bbb - 1)` address: 0x06000007 value: 0x0708

Struct Typedef (___/10)

Write the code below to make a struct with three integer fields "**bar**", "**baz**", and "**qux**" and typedef it to be of type "**FOO**".

```
typedef struct foo
{
    int bar, baz, qux;
} FOO;
```

Macro (___/10)

Define a macro called PACK that will take in **two shorts** and pack them into an **int**. PACK should respect order of operations so PACK(1 + 2, 2 + 3) should be the same as PACK(3, 5). The size of a short is **two bytes** and the size of an int is **four bytes**.

```
#define PACK(a, b) (((a)<<16) | (b))
```

32-bit integer, 16-bit short

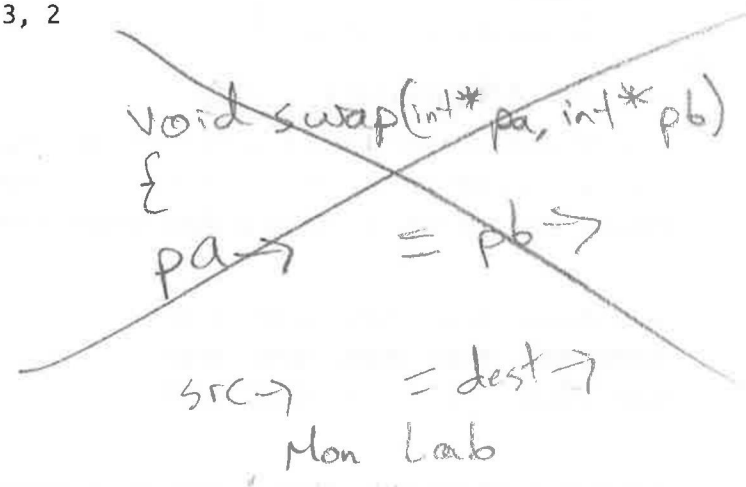
Swap (___/25)

Write a void function called swap that can switch the value of two **integers**. After the function is finished the values of the two integers should remain swapped. The parameters to your function may be of any type necessary to make your function work. Below you will see how we will call swap, the parameters are blank, after you write swap fill in the parameters correctly.

double pointer?

```
int main() {
    int a = 2;
    int b = 3;
    swap(&a, &b);     // Fill this in once you have written swap
    printf("%d, %d\n", a, b); // Should print: 3, 2
    return 0;
}
```

```
void swap(int *pa, int *pb)
{
    int temp = *pa;
    *pa = *pb;
    *pb = temp;
}
```



Short Answer (___/10)

1) The **static** keyword can be used in three different contexts in C. What are the three different contexts and what does the keyword mean in each of these contexts?

1. For a variable inside a function: Makes that variable visible to all calls of that function, so two calls to that function that each incremented by 1 would make it = 2.
2. For a global variable declaration: Makes that variable visible only within the file it is declared in.
3. For a function declaration: Like #2, restricts visibility of