COP 3275 Programming Using C  

Name: _______________________

July 10th, 2015 Quiz 6

**Total Score _______.** Your work is to be done individually. The quiz is worth 50 points and it has 8 questions. Unless a problem directly instructs you differently, there are no known errors within this document. All programming solutions will be implemented in the C programming language. You only need to write the fragment of code that directly answers each question. Unless otherwise specified you do not have to write a full program for any of the questions.

1. [15 pts] Assuming they all are part of the same function, briefly explain what the following pieces of code related to pointers do. (3pts each)

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int *ptr = NULL;</code></td>
<td>Creates a pointer to an integer and assigns NULL to it. (It is pointing to nothing)</td>
</tr>
</tbody>
</table>
| `int i = 10;  
  ptr = &i;` | Creates an integer variable and makes the ptr pointer point to the memory address where i is stored. |
| `ptr = (int*)malloc(sizeof(int));` | Dynamically allocates a new integer and makes ptr point to the memory location where that integer is stored. |
| `printf("%p\n",ptr);` | Prints the memory location that ptr is pointing to. |
| `printf("%i\n",*ptr);` | Prints the value stored in the memory location that ptr is pointing to. |

2. [5pts] (Free credit – just answer anything) Draw a picture that represents your attitude towards debugging your C programming assignments.

![Minion Debugging](image_url)
3. [15pts] Write a search function that given a single character and a character string returns the index at which that character first appears on the string. The function should return -1 if the character doesn’t exist in the string.

```c
int search(char c, char *s) {
    int i = 0;
    while(s[i] != '\0' && s[i] != c) {
        i++;
    }
    if(s[i] == '\0'){
        return -1;
    }else {
        return i;
    }
}
```

4. [15 pts] Based on the linked list we have been working on. Write a function that given a string returns a pointer to the struct student that contains that name. You may use strcmp defined in string.h to compare if two strings are equal. If the name is not found, the function should return NULL.

```c
struct student * search(struct student *head, char* name) {
    if(head == NULL) return NULL;
    if(strcmp(head->name, name) == 0) {
        return head;
    }else {
        return search(head->next, name);
    }
}
```

//There is also a solution that uses loops which is perfectly correct too.