CSSE 132 – Introduction to Systems Programming
Rose-Hulman Institute of Technology

Quiz 14 Dynamic Memory Allocation

Name (Print):_____ RHIT Username:_____

1. Why do we need the **heap** in addition to the **stack**?

   The variables on the stack become [ ].

   To overcome this, we create [ ] memory space on the heap, which remains [ ].

2. In order to allocate space on the heap, we need to call the function [ ]

   - The function takes an [ ] as input, indicating the number of bytes to allocate on the heap.
   - It returns [ ] of the newly allocated space on the heap.

3. In order to de-allocate/release space previously created on the heap, we need to call the function [ ]

   The function takes a [ ] as input, which is the address of a previously malloc-ed memory space.

4. Complete the function below that creates an `int` array on the heap with 10 entries, and assign each entry with value $0, 1, .., 9$. The function returns the address of the newly created array.

```
1  int* alloc_array(){
2    _________ array = malloc(_______ * _______);
3    // number of entries * size of each entry
4    for(int i = 0; i < 10; i++){
5      ____________ = i;
6    }
7    return ____________;
8  }
```

5. Important note of using of `sizeof()`:

   Use `sizeof()` *only to get the size of a certain* [ ].

   Never use it to measure [ ].

6. What happens if the following code is executed?

```
1 char* str = malloc(128);
2 str = malloc(128);
3 free(str);
```

   The [ ] is lost, thus causing a [ ].

7. Read the following code and estimate the memory leakage, assuming the code runs on a 64-bit machine.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct test{
5    char* name;
6    int value;
7  };
8  int main() {
9    struct test* tp;
10   for(int i = 0;i < 4; i++){
11     tp = malloc(i * sizeof(struct test));
12   }
13   free(tp);
14   return 0;
15 }
```

8. **Reflection:** What is one thing from this topic that you found interesting, OR what is one question you are now curious to learn more about?