

**COP 3502 Section 1 Quiz #1 - Dynamic Memory Allocation 9/3/2025**

Last Name: \_\_\_\_\_, First Name: \_\_\_\_\_

**Circle Lab Day/Time**

**11-T 8:30 am**

**12-T 9:30 am**

**13-T 10:30**

**14-R 8:30**

**15-R 9:30**

1) (10 pts) Complete the function below, so that it takes in integers **min**, **max** and **len**, and creates an array of size **len**, filling each index with a randomly generated integer in between **min** and **max**, inclusive, and returns a pointer to the newly dynamically allocated array. Assume **max-min** is less than 32767. Note: Documentation for C functions is included at the end of the quiz.

```
int* makeRandomArray(int min, int max, int len) {
```

```
}
```

2) (5 pts) Assuming that an integer is stored in 4 bytes, a char is stored in 1 byte, a double is stored in 8 bytes, and each pointer, regardless of type is 8 bytes, how much memory, in bytes, is dynamically allocated in the following segment of code? **Note: On some compilers extra memory is allocated for structs to ease addressing, but for the purposes of this question, assume that is not the case.**

```
typedef struct combo {
    int x;
    double y;
    char* name;
} combo;

combo** arr = calloc(1000000, sizeof(combo*));
for (int i=0; i<1000000; i++) {
    arr[i] = malloc(sizeof(combo));
    arr[i]->name = calloc(10, sizeof(char));
}
```

---

3) (5 pts) Write a segment of code to free the memory allocated in question #2. Note: The order in which you do things is critical here.

4) (15 pts) Included below is a student struct. For this question, complete a function that takes in an array of pointers to student, **studList**, the length of the array, **len**, and a double, **minGPA**. The function should find each of the students in **studList** with a GPA of at least **minGPA**, and return a dynamically allocated array of pointers to these students. Note that no new student structs should have memory allocated for them, rather, the array returned should just have pointers that point to the same structs that the pointers in **studList** point to. The array returned should be exactly the correct size. **In order to receive full credit, please use the realloc function appropriately before returning the pointer.**

```
typedef struct student {  
    char firstName[20];  
    char lastName[20];  
    int UCFID;  
    double GPA;  
} student;
```

```
student** getStudents(student** studList, int len, double minGPA) {
```

```
}
```

5) (12 pts) Write a segment of code that takes in an array of strings, **words**, the length of that array, **numWords**, determines the longest string in the array, and dynamically allocates memory for a new string and copies the contents of the longest string in the array into this new string and returns a pointer to this new string. **You are guaranteed that there will be a unique longest string.**

```
char* getLongestString(char** words, int numWords) {
```

```
}
```

6) (3 pts) What adult beverage is served at World of Beer? \_\_\_\_\_

### **C Language Documentation**

```
// Allocates size bytes and returns a pointer to the beginning  
// of the block of memory allocated.  
void* malloc (size_t size);
```

```
// Allocates an array of nitems, each which is size bytes  
// big, sets all bits to 0 and returns a pointer to the  
// beginning of the block of memory allocated.  
void* calloc(size_t nitems, size_t size);
```

```
// Attempts to resize the memory block pointed to by ptr to be  
// size bytes and returns a pointer to the beginning of the  
// block of memory allocated.  
void* realloc(void* ptr, size_t size);
```

```
// Deallocates memory pointed to by ptr.
free(void* ptr);

// Returns a pseudo-random number in the range of 0 to
// RAND_MAX (usually 32767).
int rand(void);

// Returns the length of the string pointed to by str.
int strlen(const char* str);
```

**Scratch Area – Please Clearly Mark Any Work You Would Like Graded.**