

16.216: ECE Application Programming

Spring 2013

Lecture 32: Key Questions April 24, 2013

1. Explain how dynamic memory allocation is useful.
2. Explain the `malloc()` function.
3. Explain the use of type casting, and why it is necessary with the allocation functions.

4. Explain the `calloc()` function.

5. Explain the `realloc()` function.

6. Explain the use of the `free()` function and why it is necessary.

7. Explain how dynamic memory allocation can be used with arrays.

8. What does the following program print? What is the state of the array after each allocation function call?

```
void main() {
    int *arr;
    int n, i;

    n = 7;
    arr = (int *)calloc(n, sizeof(int));
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    n = 3;
    arr = (int *)realloc(arr, n * sizeof(int));
    for (i = 0; i < n; i++) {
        arr[i] = i * i;
        printf("%d ", arr[i]);
    }

    n = 6;
    arr = (int *)realloc(arr, n * sizeof(int));
    for (i = 0; i < n; i++) {
        arr[i] = 10 - i;
        printf("%d ", arr[i]);
    }
}
```

9. Explain the following common pitfalls of dynamic memory allocation, as well as their solutions:
 - a. Memory leaks
 - b. Dangling pointers