



3. Explain the `calloc()` function.

4. Explain the `realloc()` function.

5. Explain how `free()` is used to deallocate memory.

6. **Example:** What does the following program print?

```
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]);
    }

    free(arr);
}
```

7. What are the common pitfalls of dynamic memory allocation?