- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

# Homework 24

NOTE: You are not required to hand in the following exercises, but you are strongly encouraged to complete them to strengthen your understanding of the concepts covered in class.

1. What will be the output when the following function is called?

```
void foo() {
    int x = 10;
    if (x > 0) {
        printf("%d ", x);
        x--;
        foo();
    }
}
```

2. What will be the output when the following function is called?

```
void foo() {
    static int x = 10;
    if (x > 0) {
        printf("%d ", x);
        x--;
        foo();
    }
}
```

3. Consider the following function intended to return the address of a block of *n* bytes:

```
char *getspace(int n) {
   char arr[n];
   return arr;
}
```

   A. What's wrong with the function?
   B. How would you fix the problem using `malloc`?

4. Consider the following declaration for a single node in the linked list of integers:

```
struct Node {
    int data;
    struct Node *next; // Pointer to the next node in the list
};
```

Implement the following functions:

```
/**
 * Creates a new linked list node with 'data' using malloc and
 * returns a pointer to the newly allocated node, or NULL if
 * malloc failed.
 */
struct Node *createNode(int data);

/**
 * Prints all elements in the linked list pointed by 'head'
 */
void printList(struct Node *head);

/**
 * Frees all memory allocated for the linked list pointed by 'head'
 */
void freeList(struct Node *head);

/**
 * Inserts a new node with 'data' at the head of the list
 */
void insertNode(struct Node *head, int data);
```

Note that if `malloc` fails to allocate the requested memory (for example, due to insufficient available memory), it returns `NULL`. It is essential to always check for a `NULL` return value after calling `malloc` to handle memory allocation failures safely and prevent program crashes or undefined behavior.

- **Welcome: Sean**
  - LogOut