CS 100 Exam Three – Coding – Spring 2017

- Create a directory called exam3 using mkdir exam3 and move into that directory with cd exam3 to complete the two programs shown below.
- The skeleton code for each problem (shown below) can be downloaded from the course Blackboard site.
- You are not allowed to use the Internet while coding. You can log into the cs-intro server to test your programs.

1. Name this program **one.c** – This program reads in a matrix of integers. It then calls **hasDups** to see if there are any duplicates in the matrix (a given value occurs more than once). You must write **hasDups**. **Hint**: create a helper function that takes a value **x** and the matrix and returns how many times **x** occurs in the matrix. Your **hasDups** calls this helper function for each item in your matrix.

```
#include <stdio.h>
#include <stdlib.h>
int **allocateArray(int r, int c) {
    int **array = (int **) malloc (sizeof(int *)*r);
    for (int a=0; a<r; a++)
        array[a] = (int *) malloc (sizeof(int) * c);
    return array; }
void readArray(int **array, int r, int c) {
    for (int a=0; a<r; a++)
        for (int b=0; b<c; b++)
            scanf("%d", &array[a][b]);
    return; }
void printArray(int **array, int r, int c) {
    for (int a=0; a<r; a++) {
        for (int b=0; b<c; b++)
            printf("%d ", array[a][b]);
        printf("\n"); }
    return; }
int hasDups(int **data, int, int);
// returns 1 if duplicate values exist, else 0
int main(void) {
    int **data = allocateArray(5,5);
    readArray(data, 5, 5);
    printArray(data, 5, 5);
    if ( hasDups(data, 5, 5) )
        printf("The matrix has duplicate values\n");
    else
        printf("No duplicates found\n");
    return 0;
}
```

2. Name this program two.c – This program builds a linked list of ten numbers and then prompts the user for a value. It counts how many times that value occurs in the linked list. Complete the **howMany** function.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
    int value;
    struct node *next;
} Node;

Node *add(Node *head, int data) {
    Node *newNode = (Node *) malloc(sizeof(Node));
    newNode->value = data;
    newNode->next = head;
    return newNode;
}

int howMany(Node *, int);

int main(void) {
  int a, num, count;
  Node *head = NULL;
  printf("Enter 10 values : ");
  for (a=0; a<10; a++)
      { scanf("%d", &num); head = add(head, num); }
  printf("Enter a value : ");
  scanf("%d", &num);
  count = howMany(head, num);
  printf("Value %d occurs %d times\n", num, count);
  return 0;
}
```

Submit your exam

First, compress the exam3 directory containing your two programs into a single (compressed) file.
Second, once you have a compressed file that contains your two exam3 programs, submit that file to Blackboard.