ECE 551D
Fall 2022
Test 2—Version 1

Name: NetID:

There are 5 questions, with the point values as shown below. You have 70 minutes with a total of 40 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. However, this exam is open notes, so you may use your class notes, which must be handwritten by you.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

#	**Question**	**Points**
1	Concepts	5
2	Reading Code	8
3	Pictionary	6
4	Coding 1	9
5	Coding 2	12
	Total	40

Question 1 Concepts [5 pts]

For all parts of this question, you *must* blacken the circle of the answer you choose. Remember that we can *only* see the region with the circles when grading.

1. Consider the following lines of code:

   ```
   int x = 5;
   const int * p = &x;
   int * const * q = &p;
   ```

   Which one of the following will produce a compiler error?

   (a) Modifying `x`

   (b) Modifying `p`

   (c) Modifying `*p`

   (d) Modifying `q`

   (e) None of the above

2. You expect your program to print the string "apple" for a particular test case. Sometimes it prints "apple", but sometimes it prints "applekzxn" ("apple" followed by one or more characters).

   Following the scientific method, you have observed a phenomenon and asked the question "Why does my program print the correct output sometimes but print additional characters other times?" The next step in the scientific method is to gather information. Which one of the following is the best way to gather information to formulate a hypothesis for this question?

   (a) Try a different test case that ends in a newline

   (b) Add a print statement after the line that outputs "applekzxn"

   (c) Use GDB to inspect the elements of the string

   (d) Use Valgrind to check for memory leaks

   (e) None of the above

3. The following two questions refer to this code:

```
1  #include <stdlib.h>
2
3  int main(void) {
4    size_t n = 4;
5    int * x = malloc(sizeof(n));
6    for (size_t i = 0; i < n; i++) {
7      x[i] = i;
8    }
9    free(x);
10   return EXIT_SUCCESS;
11 }
```

   Assuming `int` has size 4 bytes and `size_t` has size 8 bytes, which one of the following errors would valgrind produce?

   (a) Invalid read of size 4

   (b) Invalid read of size 8

   (c) Invalid write of size 4

   (d) Invalid write of size 8

   (e) None of the above

4. How would you fix the error?

   (a) On line 4, initialize `n` to 8

   (b) On line 5, change the argument to `malloc` to `n*sizeof(*x)`

   (c) On line 6, change the loop guard to `i < n - 1`

   (d) On line 7, change `x[i]` to `x[i-1]`

   (e) None of the above

5. In the recitation videos where Drew wrote an implementation of Blackjack, he practiced *incremental development* through:

   (a) Making minor changes to the code, testing, and repeating

   (b) Planning before starting to write code

   (c) Writing test cases before writing the program

   (d) Starting with the whole problem and breaking it down into smaller subproblems as he went

   (e) None of the above

Question 2 Reading Code [8 pts]

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

char * g(char * p, char ** q){
  q++;
  char * temp = *q;
  *p = *(temp+1);
  *q = p;
  return temp;
}

int main(void) {
  char c1[] = "red";
  char c2[] = "green";
  char c3[] = "blue";
  char * carr[] = {c1, c2, c3};
  char ** q = carr + 1;
  *(q-1) = g(carr[0], q);
  printf("%s\n", carr[0]);
  printf("%s\n", carr[1]);
  printf("%s\n", carr[2]);
  return EXIT_SUCCESS;
}
```

Execute the code by hand and write the output.

Your output should be 3 lines long. Please write each line where indicated below:

Output line 1

Output line 2

Output line 3

Question 3 Pictionary [6 pts]

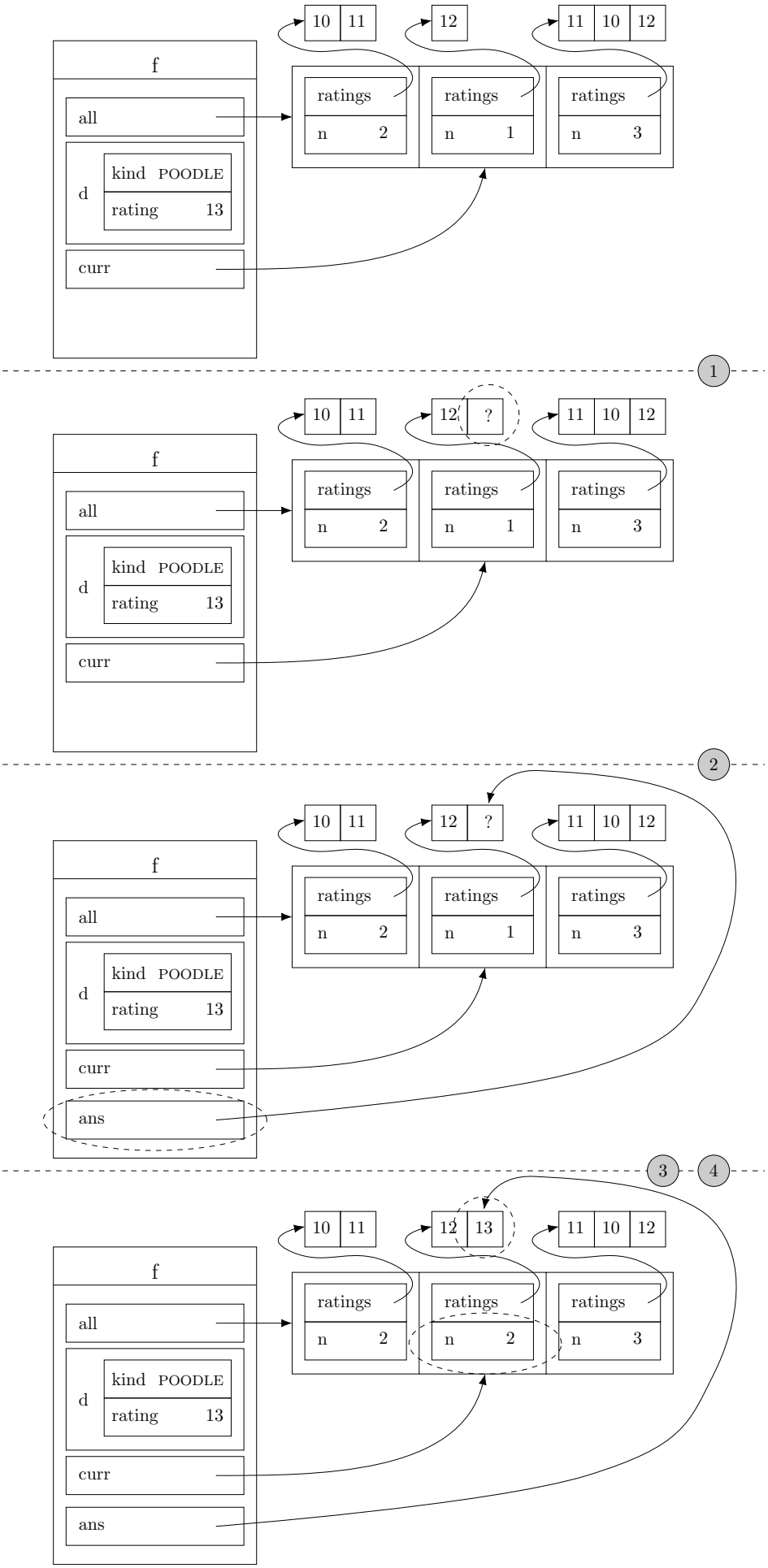
There is a Twitter handle @weratedogs that rates images of dogs on a scale of 10 (but the score always exceeds 10). Imagine you are writing the code for a function `f` that takes an array of `ratings_arr_t`s, and adds the current dog's rating to the corresponding ratings array. Note that the dog breed type is an enum, which is just a number, so we can index the array `all` with the dog's kind.

The types and provided code for `f` is given:

```
enum breed_tag {
  PITT_BULL, POODLE, MUTT
};
typedef enum breed_tag breed_t;

struct dog_tag {
  breed_t kind;
  unsigned rating;
};
typedef struct dog_tag dog_t;

struct ratings_arr_tag {
  unsigned * ratings;
  size_t n;
};
typedef struct ratings_arr_tag ratings_arr_t;

unsigned * f(ratings_arr_t * all, dog_t d) {
  ratings_arr_t * curr = &all[d.kind];
  // write line 1
  // write line 2
  // write line 3
  // write line 4
  return ans;
}
```

Using the picture on the following page, write each line in `f` where indicated below. Note that line 1 transforms the state of the program from above dashed line 1 to below, line 2 transforms the picture from above dashed line 2 to below, and lines 3 and 4 transform the picture from above dashed line 3-4 to below. For full credit, your answer must be general (use variables, not hardcoded values) for the inputs to `f`.

Line 1
Line 2
Line 3
Line 4


10 11
12
11 10 12
f
all
d
kind POODLE
rating 13
curr
ratings
n 2
ratings
n 1
ratings
n 3
1
12 ?
2
ans
3
4
13

Question 4 Coding 1 [9 pts]

Write the function `countRanges` which takes an array of ints (`arr`), and its length (`len`) and computes the length of ranges of consecutive copies of the same number. The length of each consecutive range is placed in a `ranges_t` array, which is defined as

```
struct ranges_tag {
    size_t * counts;
    size_t num_counts;
};
typedef struct ranges_tag ranges_t;
```

For example, if your function were given an input of: `{ 5, 5, 5, 8, 8, 1, 0, 0, 5 }` It would return the a `ranges_t *` whose `num_counts` field is 5, and whose `counts` field points at an array with values `{3, 2, 1, 2, 1}`. These values are the answer as there are 3 of the same number in a row (namely 5), then 2 of the same number in a row (namely 8), then a single 1, and so on.

For this problem you may assume that `malloc` and `realloc` never fail.

Please answer on the next page

```
ranges_t * countRanges(int * arr, size_t len) {

}
```

Question 5 Coding 2 [12 pts]

For this problem, you are going to write a program using the function you wrote in the previous problem (and assume it works correctly). The program should take one command line argument, the name of a file. The program will read data from the specified file, and *for each line read*:

1. Call the `lineToArray` function that we have provided.
2. Pass the result of `lineToArray` to your `countRanges` function from the previous question.
3. Use our `printRanges` function to print the values in the `ranges_t *` data structure returned from `countRanges`.

You must also be sure to free any memory and clean up any other resources you need to in the appropriate places.

Note you should NOT write `lineToArray` nor `printRanges`. You should assume we have written them already, that they work correctly, and that they have the following declarations:

```
//prints all data in ranges
void printRanges(range_t * ranges);

struct data_tag {
  int * data;
  size_t n; //length of data
};
typedef struct data_tag data_t;

//takes a line and returns a data_t with the data from the
//line parsed to an array of integers and the length of the array
data_t lineToArray(const char * line);
```

You may assume that `fopen`, `fclose`, `malloc`, and `realloc` all succeed. You may also assume that the correct number of command line arguments are passed and that the data in the file is the correct format.

You may NOT assume the length of each line.

Please answer on the next page

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

ECE 551D
Fall 2022
Test 2—Version 2

Name: NetID:

There are 5 questions, with the point values as shown below. You have 70 minutes with a total of 40 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. However, this exam is open notes, so you may use your class notes, which must be handwritten by you.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

#	**Question**	**Points**
1	Concepts	5
2	Reading Code	8
3	Pictionary	6
4	Coding 1	9
5	Coding 2	12
	Total	40

Question 1 Concepts [5 pts]

For all parts of this question, you *must* blacken the circle of the answer you choose. Remember that we can *only* see the region with the circles when grading.

1. Consider the following lines of code:

   ```
   int x = 5;
   const int * p = &x;
   int * const * q = &p;
   ```

   Which one of the following will produce a compiler error?

   (a) Modifying `x`

   (b) Modifying `p`

   (c) Modifying `q`

   (d) Modifying `*q`

   (e) None of the above

2. You expect your program to print the string "apple" for a particular test case. Sometimes it prints "apple", but sometimes it prints "applekzxn" ("apple" followed by one or more characters).

   Following the scientific method, you have observed a phenomenon and asked the question "Why does my program print the correct output sometimes but print additional characters other times?" The next step in the scientific method is to gather information. Which one of the following is the best way to gather information to formulate a hypothesis for this question?

   (a) Try a different test case that ends in a newline

   (b) Add a print statement after the line that outputs "applekzxn"

   (c) Use GDB to inspect the elements of the string

   (d) Use Valgrind to check for memory leaks

   (e) None of the above

3. The following two questions refer to this code:

```
1  #include <stdlib.h>
2
3  int main(void) {
4    int n = 4;
5    double * x = malloc(sizeof(n));
6    for (int i = 0; i < n; i++) {
7      x[i] = i;
8    }
9    free(x);
10   return EXIT_SUCCESS;
11 }
```

   Assuming `int` has size 4 bytes and `double` has size 8 bytes, which one of the following errors would valgrind produce?

   (a) Invalid read of size 4

   (b) Invalid read of size 8

   (c) Invalid write of size 4

   (d) Invalid write of size 8

   (e) None of the above

4. How would you fix the error?

   (a) On line 4, initialize `n` to 8

   (b) On line 5, change the argument to `malloc` to `n*sizeof(*x)`

   (c) On line 6, change the loop guard to `i < n - 1`

   (d) On line 7, change `x[i]` to `x[i-1]`

   (e) None of the above

5. In the recitation videos where Drew wrote an implementation of Blackjack, he practiced *incremental development* through:

   (a) Making minor changes to the code, testing, and repeating

   (b) Planning before starting to write code

   (c) Writing test cases before writing the program

   (d) Starting with the whole problem and breaking it down into smaller subproblems as he went

   (e) None of the above

Question 2 Reading Code [8 pts]

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

char * g(char * p, char ** q){
  q--;
  char * temp = *q;
  *temp = *(p+1);
  *q = p;
  return temp;
}

int main(void) {
  char c1[] = "red";
  char c2[] = "green";
  char c3[] = "blue";
  char * carr[] = {c1, c2, c3};
  char ** q = carr + 2;
  *q = g(carr[2], q);
  printf("%s\n", carr[0]);
  printf("%s\n", carr[1]);
  printf("%s\n", carr[2]);
  return EXIT_SUCCESS;
}
```

Execute the code by hand and write the output.

Your output should be 3 lines long. Please write each line where indicated below:

Output line 1

Output line 2

Output line 3

Question 3 Pictionary [6 pts]

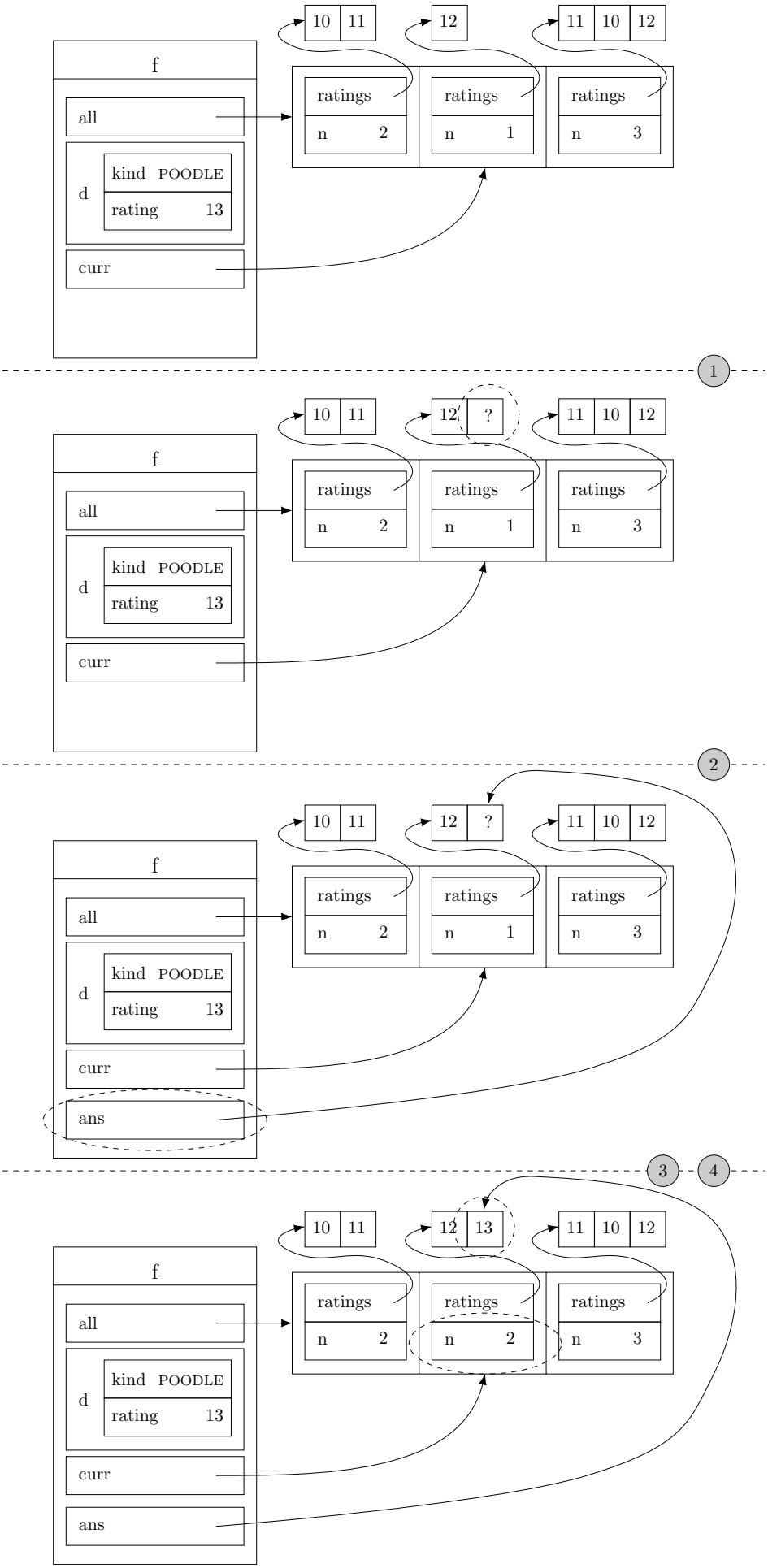
There is a Twitter handle @weratedogs that rates images of dogs on a scale of 10 (but the score always exceeds 10). Imagine you are writing the code for a function `f` that takes an array of `ratings_arr_t`s, and adds the current dog's rating to the corresponding ratings array. Note that the dog breed type is an enum, which is just a number, so we can index the array `all` with the dog's kind.

The types and provided code for `f` is given:

```
enum breed_tag {
  PITT_BULL, POODLE, MUTT
};
typedef enum breed_tag breed_t;

struct dog_tag {
  breed_t kind;
  unsigned rating;
};
typedef struct dog_tag dog_t;

struct ratings_arr_tag {
  unsigned * ratings;
  size_t n;
};
typedef struct ratings_arr_tag ratings_arr_t;

unsigned * f(ratings_arr_t * all, dog_t d) {
  ratings_arr_t * curr = &all[d.kind];
  // write line 1
  // write line 2
  // write line 3
  // write line 4
  return ans;
}
```

Using the picture on the following page, write each line in `f` where indicated below. Note that line 1 transforms the state of the program from above dashed line 1 to below, line 2 transforms the picture from above dashed line 2 to below, and lines 3 and 4 transform the picture from above dashed line 3-4 to below. For full credit, your answer must be general (use variables, not hardcoded values) for the inputs to `f`.

Line 1
Line 2
Line 3
Line 4


10 11
12
11 10 12
f
all
d
kind POODLE
rating 13
curr
ratings
n 2
ratings
n 1
ratings
n 3
1
12 ?
2
ans
3
4
13

Question 4 Coding 1 [9 pts]

Write the function `countRanges` which takes a string (`str`) and computes the length of ranges of consecutive copies of the same character. The length of each consecutive range is placed in a `ranges_t` array, which is defined as

```
struct ranges_tag {
    size_t * counts;
    size_t num_counts;
};
typedef struct ranges_tag ranges_t;
```

For example, if your function were given an input of: `"cccggazzc"` It would return the a `ranges_t *` whose `num_counts` field is 5, and whose `counts` field points at an array with values {`3, 2, 1, 2, 1`}. These values are the answer as there are 3 of the same character in a row (namely c), then 2 of the same character in a row (namely g), then a single a, and so on.

For this problem you may assume that `malloc` and `realloc` never fail.

Please answer on the next page

```
ranges_t * countRanges(const char * str) {

}
```

Question 5 Coding 2 [12 pts]

For this problem, you are going to write a program using the function you wrote in the previous problem (and assume it works correctly). The program should take one command line argument, the name of a file. The program will read data from the specified file, and *for each line read*:

1. Call the `lowerCaseAll` function that we have provided.
2. Pass the result of `lowerCaseAll` to your `countRanges` function from the previous question.
3. Use our `printRanges` function to print the values in the `ranges_t *` data structure returned from `countRanges`.

You must also be sure to free any memory and clean up any other resources you need to in the appropriate places.

Note you should NOT write `lowerCaseAll` nor `printRanges`. You should assume we have written them already, that they work correctly, and that they have the following declarations:

```
//prints all data in ranges
void printRanges(range_t * ranges);

//modifies str in place to change all characters to lowercase.
void lowerCaseAll(char * str);
```

You may assume that `fopen`, `fclose`, `malloc`, and `realloc` all succeed. You may also assume that the correct number of command line arguments are passed.

You may NOT assume the length of each line.

Please answer on the next page

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