**Daniyal Khan**

**3765942  
  
CS-2263**

**Lab 4**

**Exercise 1:**

#*include* <stdio.h>

int *areDistinct*(int \*arr, int len);

#*define* *MAX\_SIZE* 1000

int *main*(int argc, char \*\*argv) {

FILE \*fptr = *fopen*(argv[1], "r");

*if* (fptr == *NULL*) {

*printf*("Could not open file.\n");

*return* 1;

}

int arr[*MAX\_SIZE*];

int i = 0;

*while* (*fscanf*(fptr, "%d", &arr[i]) == 1 && i < *MAX\_SIZE*) {

i++;

}

*fclose*(fptr);

*if* (*areDistinct*(arr, i)) {

*printf*("The given file has distinct numbers.\n");

} *else* {

*printf*("The given file does not have distinct numbers.\n");

}

*return* 0;

}

int *areDistinct*(int \*arr, int len) {

int ind1, ind2;

*for* (ind1 = 0; ind1 < len; ind1++) {

*for* (ind2 = ind1 + 1; ind2 < len; ind2++) {

*if* (arr[ind1] == arr[ind2]) {

*// found two elements with the same value*

*return* 0;

}

}

}

*// have not found two elements of the same value*

*return* 1;

}

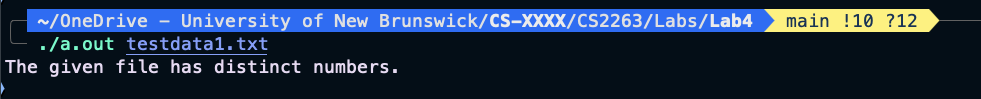
**Output:**

**No repeated numbers:**

Test data:

A screenshot of a computer

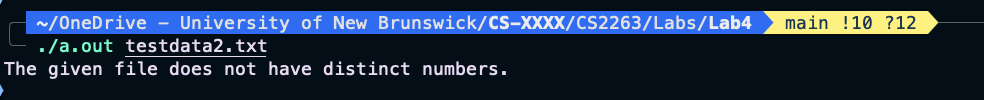
AI-generated content may be incorrect.



**Repeated numbers:**

Test data:  
A screenshot of a computer

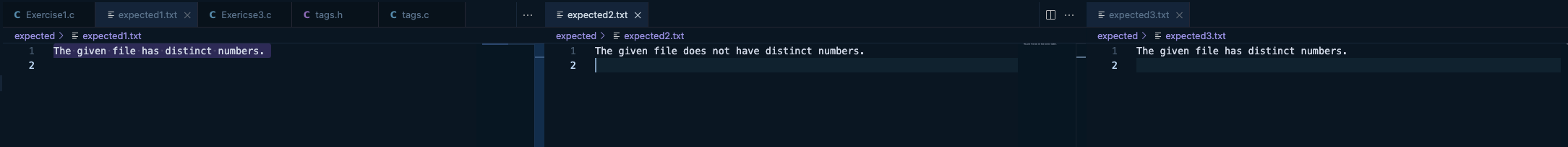
AI-generated content may be incorrect.



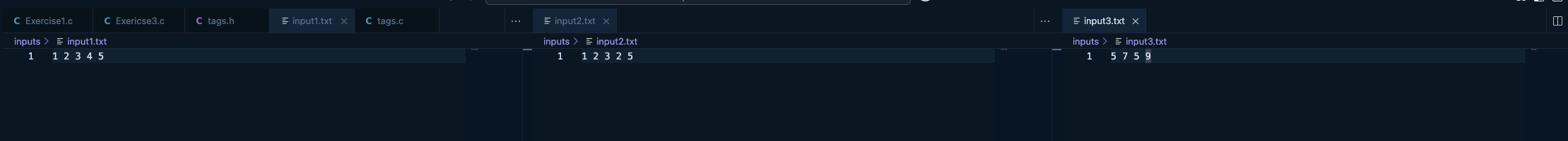
**Exercise 2:**  
  
**ls:**  
A computer screen shot of a program

AI-generated content may be incorrect.

**Contents of expected files:**



**Contents of input files:**



**Makefile:**

GCC = gcc

CFLAGS = -g -Wall -Wshadow

*prog*: Exercise1.o

$(GCC) $(CFLAGS) Exercise1.o -o prog

*Exercise1.o*: Exercise1.c

$(GCC) $(CFLAGS) -c Exercise1.c

*testall*: test1 test2 test3

*test1*: prog

./prog inputs/input1.txt > outputs/output1.txt

diff expected/expected1.txt outputs/output1.txt

*test2*: prog

./prog inputs/input2.txt > outputs/output2.txt

diff expected/expected2.txt outputs/output2.txt

*test3*: prog

./prog inputs/input3.txt > outputs/output3.txt

diff expected/expected3.txt outputs/output3.txt

*clean*:

rm -f \*.o prog outputs/\*

**Running make file to compile program:**

A screen shot of a computer

AI-generated content may be incorrect.

**ls after running it:**

A computer screen shot of a program

AI-generated content may be incorrect.

**Timestamps for .c file, .o file and the executable file**

A number on a black background

AI-generated content may be incorrect.



**Makefile to run all the tests:**

A computer screen with white text

AI-generated content may be incorrect.

**ls after it:**A screenshot of a computer program

AI-generated content may be incorrect.

**Exercise 3:**

Source code:

#*include* <stdio.h>

#*include* <stdlib.h>

#*include* "tags.h"

# *define* *N* 100000

# *define* *T* 100

void *readInput*(char \*p, int \*length);

void *countTags*(char \*arr, int length);

void *printTags*(char \*\*tags, int \*tagCounts, int tagNum);

int *main*(int argc, char \*\*argv) {

char inputArr[*N*];

int inputArrLength = 0;

*readInput*(inputArr, &inputArrLength);

*countTags*(inputArr, inputArrLength);

*putchar*('\n');

}

void *countTags*(char \*arr, int length) {

char \*tags[*T*];

int tagCounts[*T*] = {0};

int tagNum = 0;

char \*end = arr + length;

*while* (arr < end) {

*if* (\*arr == '<' && \*(arr+1) != '!' && \*(arr+1) != '/') {

*if* (!*exists*(tags, arr, tagCounts, &tagNum)) {

\*(tags+tagNum) = arr;

tagCounts[tagNum]++;

tagNum++;

}

}

arr++;

}

*printTags*(tags, tagCounts, tagNum);

}

void *printTags*(char \*\*tags, int \*tagCounts, int tagNum) {

char \*\*end = tags + tagNum;

*while* (tags < end) {

char \*tagPtr = \*tags;

*printf*("<");

*while* (\*tagPtr != '>' && \*tagPtr != ' ' && \*tagPtr != '/') {

*if* (\*tagPtr != '<') {

*printf*("%c", \*tagPtr);

}

tagPtr++;

}

*printf*(">");

*putchar*('\t');

*printf*("%i", \*tagCounts);

*putchar*('\n');

tags++;

tagCounts++;

}

}

void *readInput*(char \*p, int \*length) {

char ch;

*while* ((ch = *getchar*()) != *EOF* && \*length < *N* - 1) {

\*p = ch;

p++;

(\*length)++;

}

}

**Tags.c: (exists function)**

#*include* "tags.h"

int *exists*(char \*\*tags, char \*arr, int \*tagCounts, int \*tagNum) {

char \*\*end = tags + (\*tagNum);

*while* (tags < end) {

char \*tagPtr = \*tags;

char \*arrPtr = arr;

*while* (\*tagPtr && \*arrPtr &&

\*tagPtr != '>' && \*tagPtr != ' ' && \*tagPtr != '/' &&

\*arrPtr != '>' && \*arrPtr != ' ' && \*arrPtr != '/' &&

\*tagPtr == \*arrPtr) {

tagPtr++;

arrPtr++;

}

*if* ((\*tagPtr == '>' || \*tagPtr == ' ' || \*tagPtr == '/' ) &&

(\*arrPtr == '>' || \*arrPtr == ' ' || \*arrPtr == '/' ))

{

(\*tagCounts)++;

*return* 1;

}

tags++;

tagCounts++;

}

*return* 0;

}

**Headerfile:**

#*ifndef* *TAGS\_H*

#*define* *TAGS\_H*

int *exists*(char \*\*tags, char \*arr, int \*tagCounts, int \*tagNum);

#*endif*

**Makefile:**

GCC = gcc

CFLAGS = -g -Wall -Wshadow

*prog*: htag1.o tags.o

$(GCC) $(CFLAGS) htag1.o tags.o -o htag1

*htag1.o*: htag1.c tags.h

$(GCC) $(CFLAGS) -c htag1.c

*tags.o*: tags.c tags.h

$(GCC) $(CFLAGS) -c tags.c

*test1*: prog

./htag1 < inputs/sample.html > outputs/output1.txt

*test2*: prog

./htag1 < outputs/output1.txt > outputs/output2.txt

*compare*: test1 test2

diff outputs/output1.txt outputs/output2.txt

*tests*: compare

*clean*:

rm -f \*.o htag1 outputs/\*

**A screen shot of a computer

AI-generated content may be incorrect.Running the makefile:**

**ls after:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Comment on result from tests:**

After running make tests, the program worked but the outputs were different. The first output showed <p> counted 2 times, but in the second run it only showed 1. So the program doesn’t give the same result when run on its own output. Because the program counts the number of tags as the first input and second input had 2 and 1 tag count for <P> respectively.