Test 3 Practice

I have this declaraction in main: char line[101];

Give me the statement to create a file ptr and open the file “in.txt” for reading

File\* fptr = fopen(“in.txt”, “r”);

How would I use fgets to read in a line of text into line?

Fgets(line, 100,fptr);

How would I use fscanf to read it in?

Fscanf(fptr,”%[^\n]”, line);

What is the difference in fscanf and scanf?

Fscanf reads from a file, scanf reads from the console

How would you read an integer called myint from a file (with FILE\* fptr)?

Fscanf(fptr, “%d”, &myint);

Create a struct as a data type that has: a variable of type double called mydbl

a char array called mystr that can hold 20 characters

Call the struct: mystruct\_t

typedef struct{

double mydbl;

char mystr[21];

}mystruct\_t;

If I put this struct in a header file called mystruct.h, give me the line to include it in my source files

#include “mystruct.h”

You have this function: void funct1(mystruct\_t m) { write the print statement to print the mydbl element

printf(“%lf%, m.mydbl);

You have this function:

void funct2(mystruct\_t\* m) {

write the print statement to print the mydbl element printf(‘%lf”, m->mydbl);

Declare an array of type mystruct\_t with 10 elements. mystruct\_t m[10];

Declare/create memory for a pointer of type mystruct\_t for 10 memory elements

mystruct\_t\* m = malloc(10 \* sizeof(mystruct\_t));

Now create a linked list struct that has an element of type mystruct\_t as its datatypedef struct list\_node\_t{

mystruct\_t\*m;

list\_node\_t\* next;

}list\_node\_t;

Suppose you have 3 files: main.c funct1.c funct2.c The funct1() function calls the funct2() function.

What dependency line would you use in your makefile to compile funct1.c? funct1.o: funct1.c

Suppose you have 3 files: main.c funct1.c funct2.c

Create a Makefile that will compile these 3 source files and create an executable called pgm

You must have a separate compile for each source and a command to create the executable

pgm: main.o funct1.o funct2.o

gcc -o pgm main.o funct1.o funct2.o

main.o: main.c

gcc -c – Wall main.c

funct1.o: funct1.c

gcc -c – Wall funct1.c

funct2.o: funct2.c

gcc -c – Wall funct2.c

What does > mean in this command? ./pgm1 > file.txt

Redirects the output of pgm1 to the file, file.txt

What does < mean in this command?

./pgm2 < myfile.txt

Redirects the input of pgm1 to come from myfile.txt