

Makefile immer gegeben, die Funktionsnamen und auch die Beispielwerte in fileIO.conf oder in den linked list Elementen waren wahrscheinlich anders.

1. Gibt einen Punkt

Gegeben:

- main.c: enthält leere main Funktion, <math.h> ist inkludiert

Aufgabe:

Vervollständige main():

- Lies ersten Kommandozeilenparameter ein
- Prüfe, ob das eine Zahl zwischen 2 und 10000 ist
 - falls nein: main kann undefiniert beenden
 - falls ja: gib "1\n" aus, falls Zahl Primzahl ist und "0\n" falls nicht

2. Gibt zwei Punkte

Gegeben:

- fileIO.conf: sieht ungefähr folgendermaßen aus:

a = 15

b = 13

- fileIO.c: enthält Signatur einer Funktion: void getAB(int *a, int *b){}
- main.c: deklariert zwei ints, a und b, führt getAB(&a, &b) aus und druckt a und b auf die Kommandozeile
- fileIO.h: enthält ein paar '#include <...>', die Signatur von getAB und linkt fileIO.c und main.c.
Erklärt, dass man sich darauf verlassen kann, dass fileIO.conf genau jeweils einmal a und b enthält und nach dem '=' ein int folgt.

Aufgabe:

Vervollständige getAB():

Lies die Datei fileIO.conf und setze a und b so wie in fileIO.conf spezifiziert

3. Gibt zwei Punkte

Gegeben:

- sll.c: enthält Signatur einer Funktion: `void add_Element(struct node *list, int x)`
- main.c: Erstellt eine linked list mit ein paar Elementen, führt dann `add_Element(&first_node, 7)` aus und druckt die Werte der x der einzelnen Elemente auf die Kommandozeile, z.B. "4->3->48->7" für eine Liste mit ursprünglich drei Nodes und den entsprechenden Werten für deren x: 4, 3, 48
- sll.h: enthält Definition des Structs '`struct node {int x; struct node *next;};`' und linkt sll.c und main.c. Enthält Hinweis, dass bei letztem Element `next == NULL`.

Aufgabe:

Vervollständige `add_Element(struct node *list, int x)`:

`add_Element()` soll eine neue Node an 'list' anhängen, die den Wert 'x' enthält.

4. Gibt einen Punkt

Gegeben:

- pipe.c: enthält leere Signatur einer Funktion: `int creat_und_write(FILE *fd, *string)`
- main.c: Erstellt `int fd[2]`, den Array, der die Deskriptoren der Pipe enthalten soll, führt `creat_und_write(fd, "Hallo Welt\0")` aus und macht ein `fork()`. Im Child wird dann aus der Pipe gelesen und das gelesene auf die Kommandozeile gedruckt.
- pipe.h: linkt pipe.c und main.c, enthält Hinweis, dass man sich darauf verlassen kann, dass der gegebene String `\0`-terminiert ist.

Aufgabe:

Vervollständige `creat_und_write(FILE *fd, char *string)`:

Funktion soll Pipe erstellen, string in die Pipe schreiben und die Anzahl der geschriebenen Zeichen zurückgeben.