

Name: _____ Aufwand in h: _____

Mat.Nr: _____ Punkte: _____

Übungsgruppe: _____ korrigiert: _____

Schachtelungsanalyse für C-Programme (24 Punkte)

Bei der Erstellung eines C-Programms kann es vorkommen, dass eine öffnende oder schließende Klammer vergessen oder ein Kommentar nicht abgeschlossen wird. Dies kann zu schwer auffindbaren Syntaxfehlern führen, da der C-Compiler eine völlig andere Klammerungsstruktur annimmt und damit den Überblick verliert!

Erstellen Sie ein Programm `ccheck.c`, das C-Programme analysiert, indem es am Anfang jeder Zeile die einzelnen Schachtelungstiefen angibt, die nach dieser Zeile vorliegen. Die Zeichen `{`, `}`, `(` oder `)` bzw. `/*` und `*/` bewirken hierbei nur dann eine neue Schachtelung, wenn sie nicht in einem Kommentar oder in einer Zeichenkette angegeben sind.

Für nicht ausgeglichene Klammerung ist eine separate Meldung auszugeben!

Verwenden Sie die Standard E/A-Funktionen und lesen sich dazu das Dokument *Standard_EA_Funktionen.pdf* durch!

Der Aufruf des Programmes mit der Testdatei `tempnam.c` ergibt folgende Ausgabe:

```
1: {0} (0) /*0*/ |#include  "eighdr.h"
2: {0} (0) /*0*/ |
3: {0} (0) /*0*/ |int
4: {0} (0) /*0*/ |main(int argc, char *argv[])
5: {1} (0) /*0*/ |{
6: {1} (0) /*0*/ |    int    i;
7: {1} (0) /*0*/ |    char    *tmpdir=NULL, *praefix=NULL;
8: {1} (0) /*0*/ |
9: {1} (1) /*0*/ |    for (i=1 ;
10: {1} (1) /*0*/ |        i<argc ;
11: {2} (0) /*0*/ |        i+=2) {
12: {2} (0) /*0*/ |        if (!strcmp(argv[i], "-t") && i+1 < argc)
13: {2} (0) /*0*/ |            tmpdir = argv[i+1];
14: {2} (0) /*0*/ |        else if (!strcmp(argv[i], "-p") && i+1 < argc)
```

```

15: {2} (0) /*0*/ |          praefix = argv[i+1];
16: {2} (0) /*0*/ |          else
17: {3} (0) /*0*/ |          {
18: {3} (0) /*1*/ |              /*fehler_meld ist eine Funktion
19: {3} (0) /*0*/ |              aus der Datei eighdr.h*/
20: {3} (0) /*0*/ |              fehler_meld(FATAL, "usage:  %s [-t tmpdir] [-p praefix]", argv[0]);
21: {2} (0) /*0*/ |          }
22: {2} (0) /*0*/ |
23: {2} (0) /*0*/ |          printf("%s\n", tempnam(tmpdir, praefix));
24: {2} (0) /*0*/ |
25: {2} (0) /*0*/ |          exit(0);
26: {1} (0) /*0*/ |}

```

- Geschweifte Klammerung { } nicht ausgeglichen

Versehen Sie Ihren Quelltext mit entsprechenden Kommentaren. Geben Sie weiters Ihre Ergebnisse und alle Dateien elektronisch ab!

Allgemeine Hinweise: Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!