

Programmierung in der Bioinformatik
Wintersemester 2015
Übungen zur Vorlesung: Ausgabe am 19.10.2015

Wenn Sie daran interessiert sind sich außerhalb des Vorlesungs- und Übungsstoffes mit der Programmiersprache C auseinander zu setzen, finden Sie z.B. nützliche Informationen und Tutorials unter den folgenden Links:

- <http://www.cprogramming.com/>
- <http://c.learncodethehardway.org/>

Punktevergabe:

- Aufgabe 2.1: 3 Punkte
- Aufgabe 2.2: 2 Punkte
- Aufgabe 2.3: 1 Punkt
- Aufgabe 2.4: 1 Punkt
- Aufgabe 2.5: 2 Punkt
- Aufgabe 2.6: 1 Punkt

Welchen Editor benutzen?

Um eine neue Datei mit einem Texteditor zu erzeugen, können Sie zum Beispiel das Programm `kate` verwenden.

Um die Datei `helloworld.c` zu erstellen, tippen Sie folgenden Aufruf in ein Terminal:

```
kate helloworld.c &
```

Existiert diese Datei bereits, wird sie so geöffnet. Die Endung `.c` ist optional, hilft Ihnen aber, C-Dateien auf den ersten Blick von anderen Dateien zu unterscheiden. Außerdem wählen viele Editoren bei Verwendung von Standardendungen der jeweiligen Programmiersprache automatisch das passende Syntax-Highlighting. Mit dem `&` veranlassen Sie, dass das Programm im Hintergrund gestartet wird, und Sie das Terminal weiterhin benutzen können.

Beispiele für weitere Editoren:

Befehl	Terminal	graphisch	Syntax	
kate		✓	ja	
kwrite		✓	ja	
gedit		✓	ja	
emacs	✓		ja	von Entwicklern empfohlen ;)
xemacs		✓	ja	von Entwicklern empfohlen ;)
vim	✓		ja	von den anderen Entwicklern empfohlen ;)
gvim		✓	ja	von den anderen Entwicklern empfohlen ;)
joe	✓		nein?	
geany		✓	ja	nicht installiert

Es gibt noch viele andere Editoren. Sie sollten sich für einen Editor entscheiden, der Syntax-highlighting für Ruby und C ermöglicht (*kate* kann dies auch).

Aufgabe 2.1 Betrachten Sie folgendes Programm zur Berechnung der Fakultät:

```
include <stdio.h>                                /* use standard output routines */
#include <stdlib.h>                               /* use standard library functions */

double main(int argc, char *argv[])             /* name of function and arguments */
{                                                 /* start block of statements */
    int value, count;                           /* declare variables */

    printf("input non-negative int: ");          /* ask for input */
    if (scanf("%d",&n) != 1)                    /* try to read integer */
    {                                           /* and check if successful */
        fprintf(stderr,"incorrect input\n");    /* show error message */
        return EXIT_FAILURE;                  /* return with error code */
    }
    if (n < 0) {                               /* check if n is negative */
        fprintf(stderr,"incorrect input\n");    /* show error message */
        return EXIT_FAILURE;                  /* return with error code */
    }
    count == 1;                                /* initialize variable */
    value = 1;                                  /* initialize variable */
    do {                                        /* iterate following statements */
        value = value * count                  /* modify value */
        count = count + 1;                     /* increment count */
    }
    while (count <= n);                        /* continue while condition holds */
    printf("%d!=%d\n",n,value);                /* print result */
    return EXIT_SUCCESS;                       /* return with success */
}                                               /* end block of statements */
```

1. Sie finden das oben gezeigte Programm in Stine in der Datei `faculty_with_errors.c`. Legen Sie eine Kopie dieser Datei unter dem Namen `faculty.c` an und öffnen Sie diese in einem Editor. Öffnen Sie ein weiteres Fenster mit einer Shell und rufen den GNU C-Compiler `gcc` wie folgt auf:

```
gcc -Wall -Werror faculty.c -o faculty.x
```

Um das ausführbare Programm `faculty.x` zu erhalten, müssen Sie zunächst noch ein paar Fehler im Code beheben. Die Fehlermeldungen des Compilers können Ihnen hierbei helfen diese zu finden. Anschließend kann das Programm mit `./faculty.x` ausgeführt werden um die Fakultät einer eingegebenen Zahl zu bestimmen.

2. Das vorgestellte Programm enthält eine `do-while`-Schleife. Ersetzen Sie die Schleife durch eine `while`- bzw. `for`-Schleife, ohne die Funktionalität des Programms zu verändern.

Aufgabe 2.2 Betrachten Sie das folgende Programm

```
int num, n;
double z;
char c;
num = -1; n = 99; z = 9.99; c = '9';
num = scanf("%d %lf %c", &n, &z, &c);
printf("num = %d", num);
printf(", n = %d, z = %f, c = %c\n", n, z, c);
```

und die folgenden fünf Eingabezeilen.

```
12_4.6_Z
4.6_12_Z
12_4.6_4.6
____12_4.6____Z
1234x56.6Y
```

Bestimmen Sie für jede dieser Eingabezeilen die Ausgabe des obigen Programmes.

Aufgabe 2.3 Gegeben seien die folgenden Deklarationen:

```
double r, area, perimeter;
```

Geben Sie Ausdrücke an, die in den Variablen `area` (Fläche) und `perimeter` (Umfang) folgende Werte berechnen:

- Die Fläche eines Quadrats mit Kantenlänge `r`.
- Der Umfang eines Quadrates der Kantenlänge `r`.
- Die Fläche eines Kreises mit Radius `r`.
- Der Umfang eines Kreises mit Radius `r`.

Dabei können Sie jeweils vorraussetzen, dass `r` bereits initialisiert ist.

Aufgabe 2.4 Welche der folgenden Wörter sind gültige Bezeichner der Programmiersprache C:

- `_hello`
- `Hello37`
- `constant-value`
- `subject#1`
- `n`
- `M`
- `km_per_hour`
- `speed!`
- `sensibel$name`
- `_num_incorrect_`
- `big_long_name_many_letters`

- 12oclock

Aufgabe 2.5 Welche der folgenden Ausgaben könnten bei Verwendung des angeführten `printf`-Befehls entstehen?

```
printf("n = %3d, x = %8.4f, m = \"%-15s\"\n", n, x, m);
```

1. `n=-123, x=-123.4567, m="hello,hello"`
2. `n=1234, x=1234.567, m="hello"`
3. `n=-123, x=-1234.5670, m="hello"`
4. `n=-123, x=-1234.5670, m="hello"`
5. `n=1, x=3.1472, m="hello,hello,hello"`
6. `n=1, x=3.1472, m="hello"`
7. `n=1, x=3.1472, m="hello"`

Aufgabe 2.6 Führen Sie die Anweisungen des folgenden Programms manuell aus und geben Sie die Werte der deklarierten Variablen an.

```
int a, b, c, d, e, f, g;
a = 6;
b = a + 3*4;
c = b - b%4;
d = b/3;
e = a+b / 2;
f = (a+b/2+c)/3;
g = a-b+c-d+e-f;
```

Die Lösungen zu diesen Aufgaben werden am 02.11.2015 besprochen.