

Programmieren in C, Testat 8

1. a) Gehen Sie die Anleitung “Projects_Debugger_..._de.pdf”, s. Stud.IP, zum Anlegen von Projekten und zur Verwendung des Debuggers Schritt für Schritt am Rechner durch. Die einzelnen Programmteile sind in der Datei “ex1.zip”, s. Stud.IP, enthalten.
b) Implementieren Sie das mehrere Dateien umfassende Programm aus Abschnitt 5.7, S. 67-68, des Skriptes in einem Projekt.¹ Die einzelnen Programmteile sind in der Datei “ex2_de.zip”, s. Stud.IP, enthalten.

Machen Sie sich mit den verschiedenen Programmier Techniken vertraut, sodass Sie diese Ihrem Tutor genau(!) erklären können. Nutzen Sie dabei den Debugger für eine schrittweise Programmausführung und um sich Variablenwerte anzeigen zu lassen.

2. a) Legen Sie ein neues Projekt an und fügen Sie diesem ein Modul `vector_functions.c` hinzu. Implementieren Sie in diesem folgende Funktionen für Vektoren $a, b \in \mathbb{R}^3$:

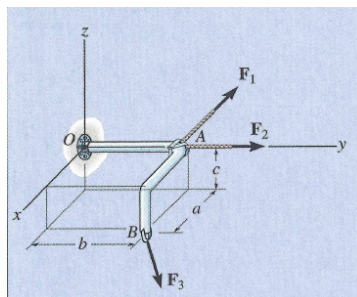
- `vecsum` : Berechnung der Vektorsumme $c = a + b$
- `crossprod` : Berechnung des Kreuzproduktes $c = a \times b$
- `dotprod` : Berechnung des Skalarproduktes $s = a \cdot b = a_1b_1 + a_2b_2 + a_3b_3$
- `vecnorm` : Berechnung der euklidischen Länge $\ell = \sqrt{a \cdot a}$

Überlegen Sie sich zunächst geeignete Signaturen der Funktionen, d.h., Anzahl und Datentypen der Eingabeparameter und Rückgabedatentypen. Schreiben Sie die Funktionsköpfe in eine entsprechende Header-Datei `vector_functions.h`.

Hinweis: Skript Abschnitt 5.2 Funktionen und Arrays.

- b) An einem starr mit einer Wand verbundenem Stab greifen n Kräfte mit Orts- und Kraftvektoren $r_i, F_i \in \mathbb{R}^3$ an, $i = 1, \dots, n$. Der Befestigungspunkt des Stabes ist als Ursprung des Koordinatensystems gewählt. Schreiben Sie ein Hauptprogramm (Modul `main.c`), das mit den Funktionen aus a) die resultierende Kraft $F_R \in \mathbb{R}^3$, das resultierende Drehmoment $M_R \in \mathbb{R}^3$, sowie deren Beträge berechnet und ausgibt.² Die Anzahl n der Kräfte soll variabel sein und zu Beginn von der Tastatur eingelesen werden. Anschließend sollen alle r_i und F_i eingegeben werden.

Beispiel: $n = 3$, $F_1 = (-60, 40, 20)$ N, $F_2 = (0, 50, 0)$ N, $F_3 = (80, 40, -30)$ N,
 $a = 4$ m, $b = 5$ m, $c = 2$ m, r_1, r_2, r_3 sind durch folgendes Bild gegeben:



Lösung: $F_R = (20, 130, -10)$ N, $|F_R| \approx 131.91$ N,
 $M_R = (30, -40, 60)$ Nm, $|M_R| \approx 78.10$ Nm

¹Erinnerung: Zur Ausgabe von Werten des Datentyps 'double' mit der printf-Funktion bei Code::Blocks unter Windows wird das Formatierungssymbol `%f` anstatt `%lf` verwendet.

²Die Lagerreaktionen des sechswertigen Festlagers sind dann $-F_R$ und $-M_R$.

English translation:

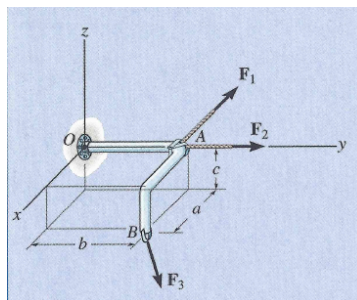
1. a) Go step by step through the tutorial “Projects_Debugger_..._en.pdf”, see Stud.IP. It describes how to create projects and how the so-called debugger is used.
b) Create a project that implements the multi-file program of Section 5.7, pp. 65-67.³ The program parts are contained in the file “ex2_en.zip”, see Stud.IP.
Become familiar with the distinct techniques and phenomena that are presented in the code so that you can explain them to your tutor in detail! Use the debugger for executing the program step by step and for checking contents of variables.
2. a) Create a new project and add a new module `vector_functions.c`. In this module, implement the following functions for vectors $a, b \in \mathbb{R}^3$:
 - `vecsum` : Computation of the vector sum $c = a + b$
 - `crossprod` : Computation of the cross product $c = a \times b$
 - `dotprod` : Computation of the dot product $s = a \cdot b = a_1b_1 + a_2b_2 + a_3b_3$
 - `vecnorm` : Computation of the Euclidean length $\ell = \sqrt{a \cdot a}$

First, consider suitable signatures for these functions, i.e., number and data types of input arguments and data types of return values. Write the function headers to a corresponding header file `vector_functions.h`.

Hint: Script section 5.2 Functions and arrays.

- b) A number of n forces acts on a rigid bar that is rigidly connected to a wall. The force-exertion and force vectors are denoted by $r_i, F_i \in \mathbb{R}^3$, respectively, $i = 1, \dots, n$. The attachment point of the bar is taken as the origin of the coordinate system. Write a main program (module `main.c`) that uses the functions from a) to compute the resulting force $F_R \in \mathbb{R}^3$, the resulting torque $M_R \in \mathbb{R}^3$, and their Euclidean lengths $|F_R|$ and $|M_R|$.⁴ The number n of the forces is variable and shall be read from keyboard at the beginning. Afterwards, all r_i and F_i are entered.

Example: $n = 3$, $F_1 = (-60, 40, 20)$ N, $F_2 = (0, 50, 0)$ N, $F_3 = (80, 40, -30)$ N,
 $a = 4$ m, $b = 5$ m, $c = 2$ m, r_1, r_2, r_3 are given by the following picture:



Solution: $F_R = (20, 130, -10)$ N, $|F_R| \approx 131.91$ N,
 $M_R = (30, -40, 60)$ Nm, $|M_R| \approx 78.10$ Nm

³Reminder: For printing 'double'-values with the printf function in Code::Blocks under Windows the formatting symbol `%f` instead of `%lf` is used.

⁴The bearing reaction force and torque are $-F_R$ and $-M_R$, respectively.