Übung 9

a) Implementieren Sie in C eine Funktion `float skalarprodukt(float a[], float b[], int n)`, die das Skalarprodukt zweier als Felder modellierter Vektoren aus dem $\mathbb{R}^n$ liefert. Stellen Sie beim Funktionsaufruf sicher, dass beide Felder dieselbe Länge n haben. Das Skalarprodukt zweier Felder mit jeweils n Elementen ist wie folgt definiert:

$[a_1,a_2,\ldots,a_n] \cdot [b_1,b_2,\ldots,b_n] = a_1b_1 + a_2b_2 + \cdots + a_nb_n$

Werden z.B. die Felder / Vektoren (1, 2, 3, 4) u. (4, 3, 2, 1) skalar multipliziert, ergibt dies 20.

b) Schreiben Sie ein kleines Programm zur Begrüßung des Users, das zuerst nach dessen Namen fragt. Speichern Sie letzteren in einem einfachen String mit ausreichender Feldlänge. Danach begrüßt Sie das Programm mit „Hallo“ gefolgt von dem eingegebenen Namen.

c) Schreiben Sie eine Funktion `int quad(char art[], int width)`, die den Parameter `art` dergestalt ändert, dass darin ein ASCII-Art-Quadrat der Breite `width` erzeugt wird, das mit dem Zeichen ‚X’ befüllt ist (ähnlich wie in der Abb. bei Aufgabenteil *f)* gezeigt). Der Rückgabewert soll die tatsächlich verwendete Länge des Feldes `art` sein.
Stellen Sie beim Funktionsaufruf sicher, dass das als Parameter übergebene Feld auch groß genug ist, um das gesamte Quadrat abspeichern zu können (notfalls erstmal mit Länge 1000 anlegen). Geben Sie in der aufrufenden Funktion das befüllte Feld zum Testen aus.

d) Ändern Sie Aufgabenteil *c)* so ab, dass statt des eindimensionalen Feldes `art` nun ein zweidimensionales Feld verwendet wird. Speichern Sie darin lediglich die für die geg. Breite nötigen ‚X’ ab. Diese Funktion benötigt keinen Rückgabewert, der Rückgabetyp ist daher *void*.

e) Schreiben Sie eine eigene Funktion `unsigned string_length(char str[])`, die die Länge des übergebenen Strings zurückgibt. Benutzen Sie nicht *strlen()* aus <string.h>. Denken Sie daran, dass Strings in C mit dem Character '\0' terminieren.

f) Implementieren Sie eine Funktion, die nebenstehendes ASCII-Art-Dreieck erzeugt: `void dreieck(int anz_zeilen);`
Dazu soll in der Funktion ein Array mit geeigneter, minimaler Größe angelegt und so mit passenden Zeichen (‚X’ und ‚\n’) befüllt werden, dass bei Ausgabe des Feldes auf die Konsole das Dreieck erscheint.
Das letzte Zeichen in Ihrem Feld soll ‚\0’ sein. Damit wird das Char-Array als String aufgefasst und lässt sich mit Hilfe des Platzhalters ***%s*** bequem (ohne Schleife) mit einem einzigen *printf()* ausgeben.

g) Schreiben Sie eine Funktion `bool palindrom(char str[])`, die überprüft, ob das übergebene Argument ein Palindrom ist oder nicht. Ein Palindrom ist ein String, der vorwärts wie rückwärts gelesen gleich ist. Testen Sie in *main()* insbesondere auch Sonderfälle wie den leeren String, ungerade vs. gerade Anzahl von Zeichen oder Strings der Länge 1.

h) Schreiben Sie eine Funktion `void entferne(char str[], char c)`, die das erste Vorkommen des Characters `c` im String `str` entfernt. D.h., dass das Argument `str` so abgeändert wird, dass es nach Aufruf nur den Reststring enthält. Kommt `c` nicht in `str` vor, bleibt `str` unverändert. Testen Sie insbesondere auch die Randfälle in der *main()* Funktion!