

4. Übung für Programmierpraktikum

Abgabetermin: 16. Dezember 2003

Name:

Matrikelnummer:

Punkte:

Die Übungen sind grundsätzlich allein zu machen. Gruppenarbeit ist nicht erlaubt. Abzugeben sind jeweils das sinnvoll dokumentierte Programmlisting mit Original-inputs und Original-outputs, falls angebracht für mehrere Testläufe mit unterschiedlichen Eingabedaten. Das Abgabeformat ist DIN A4. Heften Sie alle Unterlagen gemeinsam mit dem Übungsblatt zusammen !

Genereller Hinweis: Die Funktion `pow` wird in dieser Übung nicht benötigt und darf auch nicht benutzt werden.

1. (8 P) Aus den Vektoren $\underline{u} := \left\{ \frac{1}{(i+1)^2} \right\}_{i=0}^{n-1}$ und $\underline{w} := \left\{ \sin \left(\frac{4\pi \cdot (i-1)}{n-1} \right) \right\}_{i=0}^{n-1}$ soll der Vektor $\underline{z} = \{z_i\}_{i=0}^{2n-1}$ nach der Vorschrift

$$\{z_i\}_{i \text{ ungerade}} := \alpha \cdot \underline{u} + \underline{w} \quad \text{und} \quad \{z_i\}_{i \text{ gerade}} := \underline{u} + \beta \cdot \underline{w}$$

berechnet werden. Die Berechnungen der skalaren Größen $\alpha := \frac{\langle \underline{u}, \underline{w} \rangle}{\|\underline{u}\|}$ und $\beta := \frac{\langle \underline{u}, \underline{w} \rangle}{\|\underline{w}\|}$

enthalten inneres Produkt $\langle \underline{u}, \underline{w} \rangle := \sum_{i=0}^{n-1} u_i \cdot w_i$ und Euklidische Norm $\|\underline{u}\| := \sqrt{\sum_{i=0}^{n-1} u_i^2}$.

Der benötigte Speicherplatz ist dynamisch anzufordern (und zum Programmende explizit freizugeben). Greifen Sie auf die Komponenten von \underline{w} in der Pointernotation zu. Gestalten Sie die Berechnungen nach der Datengenerierung so, daß möglichst wenige Zählzyklen nötig sind. Geben Sie, neben α und β , für $n \leq 20$ alle Komponenten von \underline{z} aus und ansonsten die ersten und die letzten beiden Komponenten von \underline{z} .

Eingabedaten (n): (4), (20), (31553)

2. (8 P) Es sind n Richtungsvektoren einzulesen, wobei deren jeweilige Komponenten (x_i, y_i, z_i) ($i = 0, \dots, n$) gemeinsam gespeichert werden sollen. Der Speicherplatz für das Feld soll dynamisch angefordert werden (freigeben nicht vergessen!).

Bestimmen Sie den maximalen Winkel zwischen all diesen Richtungsvektoren ($\neq 0$) über

$$\cos \angle(\underline{a}, \underline{b}) := \frac{\langle \underline{a}, \underline{b} \rangle}{\|\underline{a}\| \cdot \|\underline{b}\|} \quad \text{mit} \quad \|\underline{a}\| := \sqrt{a_x^2 + a_y^2 + a_z^2} \quad \text{und} \quad \langle \underline{a}, \underline{b} \rangle := a_x b_x + a_y b_y + a_z b_z$$

und geben Sie diesen im Gradmaß ($\in [0, 180^\circ]$) zusammen mit den entsprechenden beiden Richtungsvektoren aus. Vermeiden Sie redundante Berechnungen!

Die Richtungsvektoren sind in den Files *input.4B-1.txt* und *input.4B-2.txt* als $(n, (x, y, z)_0, (x, y, z)_1, \dots, (x, y, z)_{n-1})$ gespeichert.

3. (8 P) Deklarieren Sie eine Struktur, welche bei Kontoumsätzen die Bankleitzahl, die Kontonummer, den Zweck und den geldmäßigen Umsatz (Zahlungsverpflichtung ist eine negative Zahl) speichert. Der Zweck ist eine Zeichenkette, welche dynamisch allokiert werden soll. Diese Kontoumsätze sollen wiederum in einem dynamischen Feld zusammengefaßt werden.

Allokieren Sie den dynamischen Speicher entsprechend der gegebenen Datensätze und deallokieren Sie diesen zum Schluß explizit. Bestimmen Sie den Kontoumsatz (einer reicht) mit der kleinsten Zahlungsverpflichtung und geben Sie diesen Datensatz aus. Geben Sie ebenfalls alle Datensätze aus, in denen die Zeichenkette „**USI**“ vorkommt. Hierzu können Sie die Funktion `strstr(s,s_usi)` aus `strings.h` nutzen, welche eine Nullpointer liefert falls der 2.String nicht im ersten enthalten ist.

Die *Eingabedaten* sind in den beiden Files `input.4C_1.txt`, `input.4C_2.txt` zu finden (Aufruf: `./a.out < input.4C_1.txt`).

Die Anordnung der Daten ist:

(Anzahl Umsätze, BLZ_0 , $Konto_0$, $Zweck_0$, $Umsatz_0$, \dots , BLZ_{n-1} , \dots , $Umsatz_{n-1}$)