

Lösung der 4. Übung – Informatik I für VIW

Fakultät Verkehrswissenschaften

Fachrichtung Verkehrsingenieurwesen

Zeitraum: 29.11. bis 10.12.2010 (WS 2010/11)

Aufgabe 1

```
#include <stdio.h>

int feld[100], i, j, sum, min, max;

int main() {
    for (i = 0; i<100; i++) {
        printf("Bitte geben Sie die %d. Zahl ein", i+1);
        scanf("%d", &(feld[i]));
    }

    sum = 0;
    for (i = 0; i<100; i++) {
        sum += feld[i];
    }
    printf("Der Durchschnitt ist %d.\n", sum/100);

    min = feld[0]; max = feld[0];
    for (i = 1; i<100; i++) {
        if (min > feld[i])
            min = feld[i];
        if (max < feld[i])
            max = feld[i];
    }
    printf("Das Minimum ist %d. ", min); printf("Das Maximum ist %d.\n", max);

    for (i = 0; i<100; i++) {
        if (feld[i] % 2 == 0) {
            printf("%d", feld[i]);
            if (i < 99)
                printf(", ");
            else
                printf(".\n");
        }
    }

    for (i = 0; i<100; i++)
```

```

    for (j = i+1; j<100; j++)
        if (feld[i] == feld[j])
            printf("Doppelter Wert eingegeben: %d.\n", feld[i]);

    return 0;
}

```

Aufgabe 2

(a)

```

double wurzel(int a, int n) {
    float x = 1.0;
    int i;
    for (i = 1; i <= n; i++)
        x = (x+a/x)/2;

    return x;
}

```

(b)

```

#include <stdio.h>

double wurzel(int a, int n) {...}

int main() {
    int a, n;
    printf("Bitte geben Sie den Wert a ein: "); scanf("%d", &a);
    printf("Bitte geben Sie den Wert n ein: "); scanf("%d", &n);
    printf("Der Naeherungswert lautet: %f", wurzel(a,n));
    return 0;
}

```

Aufgabe 3

(a)

```

int fiba(int n) {
    int prev, prevprev, current, i;

    prev = 1; prevprev = 0;

    for (int i = 2; i<=n; i++) {
        current = prev + prevprev; /*neuen Wert berechnen*/
        prevprev = prev;          /*der neue vorvorhergehende Wert

```

```

        ist der alte vorhergehende Wert*/
    prev = current;      /*der neue vorhergehende Wert
                           ist der alte aktuelle Wert*/
}

return prevprev;
}

```

(b)

```

int fibb(int n) {
    if (n < 1)
        return -1; /*Fehlerhafte Eingabe fuer n*/
    if (n == 1)
        return 0;
    if (n == 2)
        return 1;

    return fibb(n-1) + fibb(n-2);
}

```

(c) Die Funktion `fibb` ist zwar leichter zu verstehen, ist aber sehr viel langsamer in der Berechnung. Die Funktion `fiba` arbeitet sehr effizient: die Schleife in `fiba` wird für jedes n ungefähr n mal ausgeführt. Die Funktion `fibb` ist dagegen sehr ineffizient, da viele Werte mehrfach berechnet werden, wie das folgende Beispiel für den Aufruf `fibb(6)` zeigt:

$$\begin{aligned}
 \text{fib}(6) &= \text{fib}(5) + \text{fib}(4) \\
 &= (\text{fib}(4) + \text{fib}(3)) + (\text{fib}(3) + \text{fib}(2)) \\
 &= ((\text{fib}(3) + \text{fib}(2)) + (\text{fib}(2) + \text{fib}(1))) + ((\text{fib}(2) + \text{fib}(1)) + \text{fib}(2)) \\
 &= (((\text{fib}(2) + \text{fib}(1)) + \text{fib}(2)) + (\text{fib}(2) + \text{fib}(1))) \\
 &\quad + ((\text{fib}(2) + \text{fib}(1)) + \text{fib}(2)) \\
 &= ((1 + 0) + 1) + (1 + 0) + ((1 + 0) + 1) \\
 &= 5.
 \end{aligned}$$

Die Berechnung von `fib(6)` teilt sich in 5 Berechnungen von `fib(2)` und 4 Berechnungen von `fib(1)`. Diese Berechnungen müssen wiederholt ausgeführt werden, sind aber eigentlich unnötig.

Aufgabe 4 (Zusatz)

(a)

```

#include <stdio.h>

int jahr[50], monat[50], tag[50], stunde[50], minute[50], sekunde[50], i, j;

```

```

int main() {
    for (i = 0; i<50; i++) {
        printf("Geben Sie jetzt das %d. Datum ein:\n", i+1);
        printf("  Jahr: "); scanf("%d", &(jahr[i]));
        printf("  Monat: "); scanf("%d", &(monat[i]));
        printf("  Tag: "); scanf("%d", &(tag[i]));
        printf("  Stunde: "); scanf("%d", &(stunde[i]));
        printf("  Minute: "); scanf("%d", &(minute[i]));
        printf("  Sekunde: "); scanf("%d", &(sekunde[i]));
    }

    for (i = 0; i<50; i++)
        for (j = i+1; j<50; j++) {
            if (jahr[i] == jahr[j] &&
                monat[i] == monat[j] &&
                tag[i] == tag[j] &&
                stunde[i] == stunde[j] &&
                minute[i] == minute[j] &&
                sekunde[i] == sekunde[j]) {
                printf("Der %d. Eintrag und der", i+1);
                printf("%d. Eintrag sind gleich.\n", j+1);
            }
        }
    return 0;
}

```

(a)

```

#include <stdio.h>

struct {
    int jahr, monat, tag, stunde, minute, sekunde;
} zeitpunkte[50];

int i, j;

int main() {
    for (i = 0; i<50; i++) {
        printf("Geben Sie jetzt das %d. Datum ein:\n", i+1);
        printf("  Jahr: "); scanf("%d", &(zeitpunkte[i].jahr));
        printf("  Monat: "); scanf("%d", &(zeitpunkte[i].monat));
        printf("  Tag: "); scanf("%d", &(zeitpunkte[i].tag));
        printf("  Stunde: "); scanf("%d", &(zeitpunkte[i].stunde));
        printf("  Minute: "); scanf("%d", &(zeitpunkte[i].minute));
    }
}

```

```

    printf("   Sekunde: "); scanf("%d", &(zeitpunkte[i].sekunde));
}

for (i = 0; i<50; i++)
    for (j = i+1; j<50; j++) {
        if (zeitpunkte[i].jahr == zeitpunkte[j].jahr &&
            zeitpunkte[i].monat == zeitpunkte[j].monat &&
            zeitpunkte[i].tag == zeitpunkte[j].tag &&
            zeitpunkte[i].stunde == zeitpunkte[j].stunde &&
            zeitpunkte[i].minute == zeitpunkte[j].minute &&
            zeitpunkte[i].sekunde == zeitpunkte[j].sekunde) {
            printf("Der %d. Eintrag und der", i+1);
            printf("%d. Eintrag sind gleich.\n", j+1);
        }
    }
return 0;
}

```

Aufgabe 5 (Zusatz)

```

void swap(int *a, int *b) {
    int c = *a;
    *a = *b;
    *b = c;
}

```

Aufruf:

```

int main() {
    int a, b;
    swap(&a, &b);
}

```