

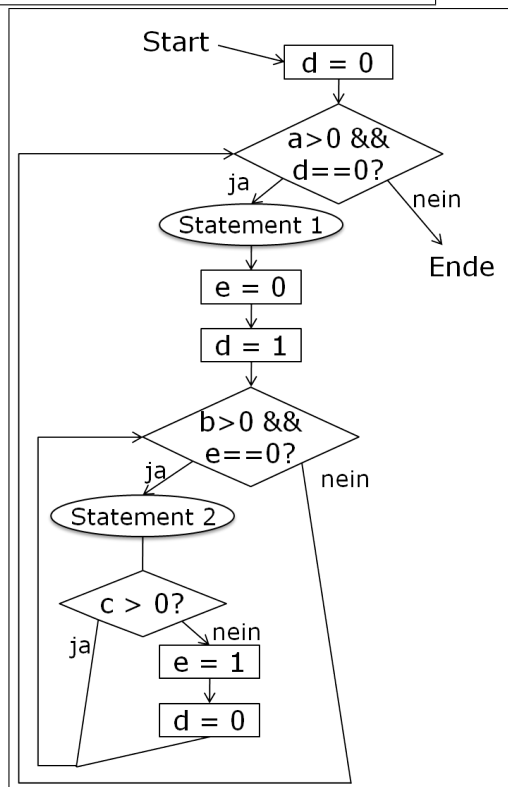
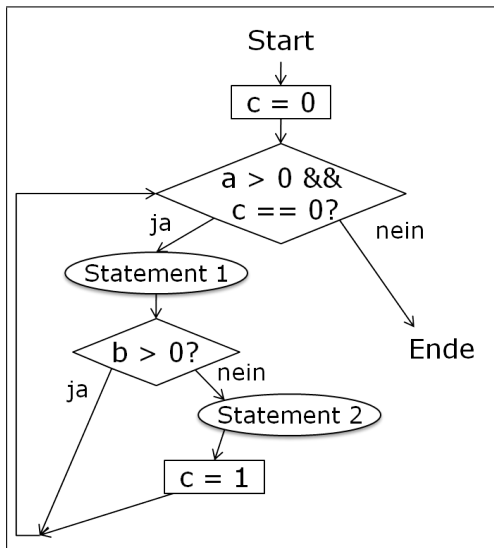
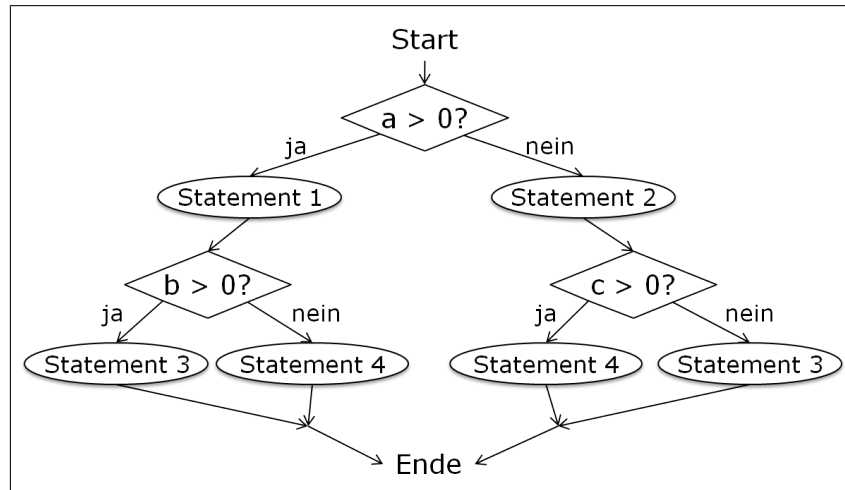
Lösung der 3. Übung – Informatik I für VIW

Fakultät Verkehrswissenschaften

Fachrichtung Verkehrsingenieurwesen

Zeitraum: 15.11. bis 26.11.2010 (WS 2010/11)

Aufgabe 1



Aufgabe 2

Programm 1

```
a++;
if (!(a<0))
{
    b+=a;
    abbruch = 0;
    while (a<c && abbruch == 0) {
        a++;
        if (a<0)
            abbruch = 1;
        else
            b+=a;
    }
}
```

Programm 2

```
i=1;
abbruch=0;
while (i<=c && abbruch == 0) {
    if (!(c<d)) {
        a++; c--;
        if (a > e)
            abbruch=1;
    }
    if (abbruch==0)
        i++;
}
```

Programm 3

```
while (i>3) {
    a-=c;
    i-=d;
}
while (a>b) {
    while (i>3) {
        a-=c;
        i-=d;
    }
}
```

Aufgabe 3

(a) 1 2 5 11 12 15

(b)

Folge 1

```
#include <stdio.h>
int a, i;

int main() {
    a = 0;
    i = 2;
    while (i>=0) {
        printf("%d ", i);
        if (i==12) a=1;
        if (a==0) i+=2;
        else i-=3;
    }
    return 0;
}
```

Folge 2

```
#include <stdio.h>
int i, j;

int main() {
    for (i=1; i<=4; i++)
        for (j=0; j<=3; j++)
            printf("%d ", i+j);
    ...
    return 0;
}
```

Folge 3

```
#include <stdio.h>
int n, i, j;

int main() {
    for (i=1; i<=6; i++) {
        n=1;
        for (j=1; j<=i; j++)
            n*=i;
    }
}
```

```

        printf("%d ",n);
    }
    return 0;
}

```

Aufgabe 4

Achtung! Wird in der Schleife im Fall, dass `feld[mitte] > key` *nicht* erfüllt ist, die Variable `anfang` nur auf `mitte` (statt `mitte+1`) gesetzt, dann kann es in manchen Fällen passieren, dass die Schleife nicht endet! Das passiert zum Beispiel, wenn `anfang==2` und `ende==3`; dann ist `mitte==2` (da beim Teilen ganzer Zahlen abgerundet wird); wenn dann der Eintrag 2 in `feld` kleiner als `key` ist, dann würde `anfang` auf `mitte` gesetzt werden und (da beide den Wert 2 haben), keine Änderung erfolgen.

Deshalb muss `anfang` auf `mitte+1` gesetzt werden (und entsprechend `ende` auf `mitte-1`). Das erfordert allerdings einen expliziten Test auf den Fall `feld[mitte] == key`, damit ein vorzeitiges Auffinden des Wertes `key` nicht vernachlässigt wird.

```

anfang = 0;
ende = laenge-1;
while (anfang <= ende) {
    mitte = (anfang + ende)/2;
    if (feld[mitte] == key)
        break;
    if (feld[mitte] > key)
        ende = mitte-1;
    else
        anfang = mitte+1;
}
if (anfang <= ende && feld[mitte] == key)
    printf("Der Eintrag \"key\" wurde in \"feld\" an Position %d gefunden.", mitte);
else
    printf("Der Eintrag \"key\" ist in \"feld\" nicht vorhanden.");

```

Aufgabe 5 (Zusatz)

```
#include <stdio.h>
int x, y, dir, raster[101][101];

int main() {
    for (x = 0; x<=100; x++)
        for (y = 0; y<=100; y++)
            raster[x][y] = 1;

    dir = 0; /*0 = Norden, 1=Osten, 2=Sueden, 3=Westen*/
    x = 50; y = 50;
    while (x<=100 && x>=0 && y<=100 && y>=0) {
        if (raster[x][y] == 1) {
            dir++;
            if (dir == 4)
                dir = 0;
            raster[x][y]=0;
        }
        else {
            dir--;
            if (dir == -1)
                dir = 3;
            raster[x][y]=1;
        }

        switch (dir) {
            case 0: y--; break;
            case 1: x++; break;
            case 2: y++; break;
            case 3: x--; break;
        }
    }
    return 0;
}
```