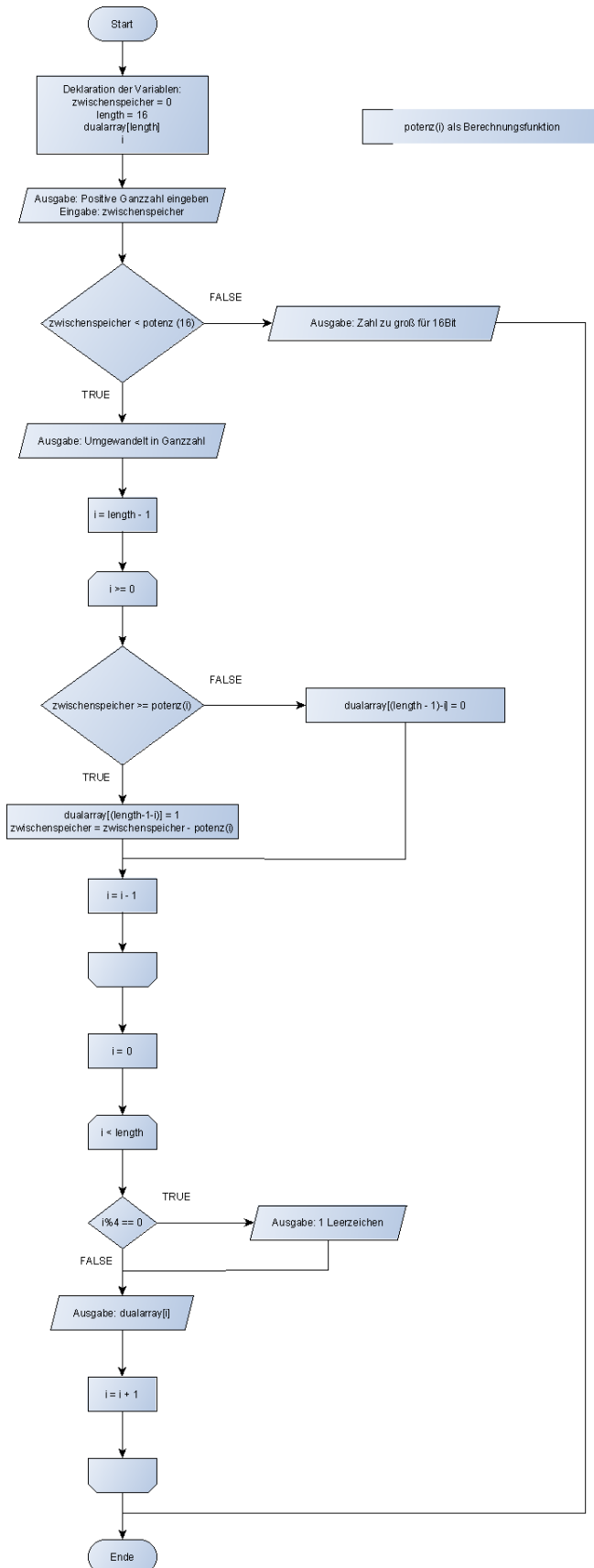


1 Beschreibung der Lösung

Das Programm rechnet eine positive ganze Zahl zur Basis 10 in eine Dualzahl (Zahl zur Basis 2) mit 16 Bit um. Zuerst wird eine Zahl eingelesen. Das Programm testet ob, die Dezimalzahl in eine Dualzahl mit 16 Bit umgerechnet werden kann (maximal: 1111 1111 1111 1111). Wenn dies nicht möglich ist, wird eine Fehlermeldung ausgegeben. Ist die Umrechnung möglich, wird getestet, ob das höchste Bit der Dualzahl (2^{15}) in die eingelesene Dezimalzahl passt. Ist dies nicht der Fall, wird in einen Array an der 1. Position der Wert „0“ gespeichert. Wenn doch, der Wert „1“ und zudem wird von der eingegebenen Zahl die höchste Potenz der Dualzahl abgezogen. Dieser Zyklus wiederholt sich in einer Schleife bis zur kleinsten Bit der Dualzahl (2^0) dabei wird die aktuelle Array-Position um eins erhöht. Nach der Berechnung werden die einzelnen Positionen des Arrays mit Hilfe einer Schleife ausgegeben.

2 Ablaufdiagramm



3 Quellcode

```
#include <stdio.h>

int potenz(int n)
{
    int zs = 2;
    int j;

    if(n == 0)
    {
        zs=1;
    }
    for(j=1;j<n;j++)
    {
        zs = zs*2;
    }
    return zs;
}

int main(void)
{
    int zwischenspeicher = 0;
    int length = 16;
    int dualarray[length];
    int i;

    printf("\n\nPositive Ganzzahl eingeben: \n");
    scanf(" %d",&zwischenspeicher);
    printf("\n");
    if(zwischenspeicher < potenz(16))
    {
        printf("Umgewandelt in Dualzahl: ");
        for(i = length-1;i>=0; i--)
        {
            if(zwischenspeicher >= potenz(i))
            {
                dualarray[(length-1)-i]=1;
                zwischenspeicher = zwischenspeicher - potenz(i);
            }
            else
            {
                dualarray[(length-1)-i]=0;
            }
            if((i+1)%4==0)
            {
                printf(" ");
            }
            printf("%d",dualarray[(length-1)-i]);
        }
    }
    else
    {
        printf("Die Zahl %d ist zu groß fuer 16Bit. Maximal %d\n!",zwischenspeicher,potenz(16));
    }
    printf("\n\n");

    return 0;
}
```

4 Testfälle

Eingabe (Dezimal)	Ausgabe (Dual)
5	0000 0000 0000 0101
70000	Programmierte Fehlermeldung
0	0000 0000 0000 0000
65535	1111 1111 1111 1111
-5	0000 0000 0000 0000 (jeder negative Zahl wird vom Programm als „Null“ gewertet.)