Die Speicherung aller Daten erfolgt im Rechner in binärer Form (mit 0 und 1). Ganze Zahlen werden als Dualzahlen dargestellt und negative Zahlen im Zweierkomplement. Zur besseren Darstellung binärer Daten werden diese häufig als Hexadezimalzahlen geschrieben. In C werden konstante Hexadezimalzahlen mit dem Präfix 0x… versehen.

Beispiele:

Dezimal	*Dualzahl*	*Hex-Zahl*
51	00110011	0x33
103	01100111	0x67
189	10111101	0xBD

Binäre Operationen sind vor allem bei der Programmierung von Mikrocontrollern von Bedeutung.
Dazu gehören beispielsweise die UND-/ODER- /XOR- und NICHT-Operationen sowie bitweises Schieben.

Beispiel für eine binäre UND-Operation:

Dezimal	*Binär*	*Hexadezimal*
103	01100111	0x67
UND 189	10111101	0x9D
37	00100101	0x25

Der Operator für eine UND-Verküpfung ist das &-Zeichen.

Beispiel für ein 2-faches Links-Schieben (entspricht einer Multiplikation mit $2^2 = 4$):

51	00110011	0x33
51<<2=204	11001100	0xCC

Der Operator für Links-Schieben ist die <<-Zeichenfolge.

Das folgende Beispiel demonstriert die dargestellten Zusammenhänge in hexadezimaler Ausgabe:

```
#include<stdio.h>
#include<conio.h>
int main()
{
     unsigned char a=0x33,b=0x67,c=0xbd;
     printf("%02x UND %02x ist %02x\n",b,c,b&c);
     printf("%02x 2 mal links Schieben ist %02x\n",a,a<<2);
     _getch();
     return(0);
}
```

Ausgabe:

```
67 UND bd ist 25
33 2 mal links Schieben ist cc
```

Der Platzhalter mit der Formatierungsangabe `"...%02x..."` innerhalb der printf()-Formatanweisung bewirkt eine zweistellige Ausgabe als Hex-Zahl mit führenden Nullen.

Die folgende Tabelle zeigt die möglichen Bit-Operationen:

Operator	*Bedeutung*
&	Bitweises UND
|	Bitweises ODER
^	Bitweises XOR
~	Bitweise Negation
<<	Linksschieben
>>	Rechtsschieben
&=	Kurzform bitweises UND mit Zuweisung
|=	Kurzform bitweises ODER mit Zuweisung
^=	Kurzform bitweises XOR mit Zuweisung
<<=	Kurzform Linksschieben mit Zuweisung
>>=	Kurzform Rechtsschieben mit Zuweisung

Übung :

Analysieren Sie das folgende Programmbeispiel und bestimmen Sie die Ausgabe bevor Sie das Programm testen!

```
#include<stdio.h>
#include<conio.h>
int main()
{
      unsigned char a=0x3d,b=0x67,c=0xa5;
      unsigned char d,e,f;
      a<<=1;
      d=b|a;
      e=d&c;
      f=(c&e)>>1;
      printf("\n%02x %02x %02x\n",d,e,f);
      _getch();
      return(0);
}
```