

u_binaer.c

```
/* Datei: u_binaer.c      Hofmann Anton
 * Demo fuer bitorientierte Operatoren
 */
```

```
#include <stdio.h>
#define BITBREITE 32
```

```
void binaer(unsigned int);
```

```
/*  main  */
```

```
/*  binaer
// konst= 1000 0000
// wenn a mit konst AND verknüpft wird,
// so ist das Ergebnis eine '1', wenn das MSB-Bit von a '1'
// sonst ist das Ergebnis '0' */
```

u_binaer.c §1

main

Funktion main()

```
int x= 1, y= 2, z= ~y;
```

```
binaer(x);
binaer(y);
binaer(z);
```

```
z= z & y;
binaer(z);
```

```
z= x | y;
binaer(z);
```

```
z= z ^ y;
binaer(z);
```

```
z= z << y;
binaer(z);
```

```
z= z >> y;
binaer(z);
```

```
binaer
// konst= 1000 0000
// wenn a mit konst AND verknüpft wird,
// so ist das Ergebnis eine '1', wenn das MSB-Bit von a '1'
// sonst ist das Ergebnis '0'
```

Funktion void binaer(unsigned int a)

```
unsigned int i, konst= 0;
```

```
konst= ????????????????; /*1 mit folgenden Nullen */
```

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

```
if ( ???????????? )
```

```
then
```

```
else
```

```
putchar ('1');
```

```
putchar ('0');
```

```
a= a ????????????; /* das naechste bit nach vorne */
```

```
putchar ('\n');
```