

*/\*Uebergabe eines Zeigers auf einen Int-Wert ans HP, der im UP vergeben wird\*/*

```
#include <stdio.h>
#include <stdlib.h>

void zahl(int *wert)
{
    int i=30;                // der Wert in i soll mal zurückgegeben werden

    printf( "%04x\n ", &i ); // zur Probe die Adresse von i ausgegeben
    printf( "%i\n"   , i );  // und den Wert von i

    *wert = i;              // dem Zeiger den Wert übergeben

    // und zurück zum Hauptprogramm
}

int main()
{
    int *p_wert;             // Einen Zeiger deklarieren
    p_wert = malloc(sizeof(int)); // und ihm dynamisch den für den Variablentyp
                                // notwendigen Speicher zuweisen

    zahl(p_wert);           // dem Zeiger den Wert aus der
                                // Funktion mitgeben

    printf( "%04x\n" , p_wert); // Und zur Kontrolle die Adresse
    printf( "%i"     , *p_wert); // und den Wert wieder ausgeben

    free(p_wert);          // den Speicher wieder freigeben

    return 1;
}
```

**/\*Uebergaben eines kompletten Strings von einem UP an ein HP\*/**

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

void neuer(char *zeichenkette)
{
    char satz[]="Aber hallo!";

    printf("%04x\n",satz);           // zur Probe die Adresse ausgeben

    strcpy( zeichenkette , satz ); // der String muss mit strcpy kopiert werden,
                                    // da zeichenkette = satz nur die Adresse von
                                    // satz in zeichenkette kopieren wird.

    printf("%04x\n",zeichenkette); // zur Sicherheit noch mal die Adresse von
    printf("%s\n",zeichenkette);   // Zeichenkette und den Inhalt ausgeben
}

int main()
{
    char p_neu[255];                // Speicherplatz für 255 bytes reservieren

    neuer(p_neu);                  //zeiger vom UP empfangen

    printf("%04x\n",p_neu);
    printf("%s",p_neu);            //hier wird der satz wieder ausgegeben werden

    return 1;
}
```