


```

case '\13':
case '\21':
    printf("\nG2 %d G0  G0 %s G0",yp,zadr);
    break;
case '\6':
case '\14':
case '\22':
    printf("\nG2 %d G0  G0 %s G0",yp,zadr);
    break;

default:
    printf("\nG4 %d G0  G2 %s G0",yp,zadr);
}
/* Ende showline */

```

```

main()
{
printf("\n\n\n\n\n\n\n");
printf("\nG4  Test der wichtigsten Bibliotheksfunktionen  G0");
printf("\n\n\nG4  Grosseze der Datentypen  G0");
printf("\n  short:          %d",sizeof(short));
printf("\n  char:           %d",sizeof(char));
printf("\n  int:            %d",sizeof(int));
printf("\n  long:          %d",sizeof(long));
printf("\n  float:         %d",sizeof(float));
printf("\n  double:        %d",sizeof(double));
printf("\n  (char *):      %d",sizeof(char *));

printf("\n\nG4  Konvertierungen  G0");
iret = atoi(" -123 ");
printf("\n Ergebnis atoi: %d",iret);
lret = atol(" +123123 ");
printf("\n Ergebnis atol: %ld",lret);
fret = atof(" -123.30e7 ");
printf("\n Ergebnis atof: %f",fret);
iret = isascii('A');
printf("\n Ergebnis isascii fuer A: %d",iret);
iret = isalpha(' ');
printf("\n Ergebnis isalpha fuer : %d",iret);

printf("\n\n G4  float und double Test in printf()  G0");
printf("\n  4.5  als float          %f",4.5);
printf("\n  4.5  scientific float    %e",4.5);
printf("\n  4.5  in g Format         %g",4.5);

printf("\n\n G4  mathematische Routinen  G0");
pi = 3.1415926;
dv = sin(45.00 * pi / 180.00);
dw = sin(89.90 * pi / 180.00);
printf("\n  pi = %g",pi);
printf("\n  sin(45) = %g          sin(89,9) = %g",dv,dw);
dv = cos(45.00 * pi / 180.00);
dw = sin(89.90 * pi / 180.00);
printf("\n  cos(45) = %g          cos(89,9) = %g",dv,dw);
dv = tan(00.10 * pi / 180.00);
dw = tan(89.90 * pi / 180.00);
dx = dw;
printf("\n  tan(00,1) = %g          tan(89,9) = %g",dv,dw);
dv = 1/(tan(00.10 * pi / 180.00));
dw = 1/(tan(89.90 * pi / 180.00));
printf("\n  cot(00,1) = %g          cot(89,9) = %g",dw,dv);
dv = ((dw * 100) / dx );

```

③

```

printf("\n Abweichung = %g Prozent ",dv);
dv = tan(45.00 * pi / 180.00);
dw = 1/(tan(45.00 * pi / 180.00));
printf("\n tan(45) = %g cot(45) = %g",dv,dw);
dv = log(15.1542623);
dw = log(dv);
printf("\n\n ln(15.5) = %g ln(ln15) = %g",dv,dw);
dx = (dw * 100) - 100;
printf("\n Abweichung = %g Prozent ",dx);
dv = log10(10.0);
printf("\n log10(10.0) = %g",dv);
dv = sqrt(2.0);
printf("\n sqrt(2.0) = %g",dv);
dv = sqrt(sqrt(122.2222));
dw = sqrt(sqrt(dv));
dv = sqrt(sqrt(dw));
dw = sqrt(sqrt(dv));
dv = ((dw * 100) / 1.01895) - 100;
printf("\n\n");
printf("\n 8te Wurzel aus 122.2222 = %g Abweichung = %g Prozent",dw,dv)
printf("\n\n");

```

```

printf("\n\n G4 Test Basis-IO G0\n");
/* mit Test, ob write per default transparent ist */
printf("\n GA Test Basis-IO G0");
printf("\n File anlegen ");
fd1 = creat("xx.test",0x01c0); /* identisch chmod 0700 */
printf("\n 36 Zeichen reinschreiben ");
iret = write(fd1,"a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,r,s\n",36);
printf("\n File schlieszen ");
close(fd1);
printf("\n und wieder oeffnen ");
fd1 = open("xx.test",0);
printf("\n\n GA Ergebnis open = G0 %d",fd1);
printf("\n Das geschriebene wieder lesen ");
iret = read(fd1,puffer,36);
printf("\n Ausgabe auf Terminal mit Basis-IO \n ");
write(1,"\n GA Ergebnis read = G0 ",27);
write(1,puffer,36);
printf("\n prüfen, ob \n zu LF+CR expandiert wurde ");
iret = read(fd1,puffer,1);
if (iret <= 0)
    printf("\n\n\t\t write-read arbeiten transparent");
else printf("\n\n\t\t G6 write-read arbeiten nicht transparent");
close(fd1);
printf("\n\n");

```

```

printf("\n\n G4 Stringroutinen G0");
printf("\n strcmp : %d",strcmp("ABC", "abc"));
printf("\n strncmp : %d",strncmp("ABC", "abcdef", 4));
printf("\n strcpy : %s",strcpy(puffer, "vwxyz"));
printf("\n strncpy : %s",strncpy(puffer, "abcdefg", 3));

```

```

cadr = strchr("ABCDEFGH", 'D');
printf("\n strchr : %lx",cadr);
cadr = strchr("ABCDEFGH", 'X');
printf("\n strchr : %lx",cadr);

```

```

printf("\n\n G4 Bildschirmfuellung Standard printf G0");
/* Zeit: 33 Sekunden fuer 200 Zeilen */

```

```

ypos = 0;
printf("\n\n");
for (n=0; n < 100; n++)
{

```

```

    if (++ypos >= 25) ypos = 1;
    showline(ypos,

```

```

"abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789");

```

```

printf("\nG4 E N D E \n\n\n\n");

```


2

```

case '\3':
case '\11':
case '\17':
case '\25':
    printf("\nG2 %d G0 G8 %s G0",yp,zadr);
    break;

case '\4':
case '\12':
case '\20':
case '\26':
    printf("\nG2 %d G0 G4 %s G0",yp,zadr);
    break;

case '\5':
case '\13':
case '\21':
    printf("\nG2 %d G0 GD %s G0",yp,zadr);
    break;

case '\6':
case '\14':
case '\22':
    printf("\nG2 %d G0 G0 %s G0",yp,zadr);
    break;

default:
    printf("\nG4 %d G0 G2 %s G0",yp,zadr);
}
/* Ende showline */

```

```

main()
{
printf("\n\n\nG4 Groesse der Datentypen G0");
printf("\n short: %d",sizeof(short));
printf("\n char: %d",sizeof(char));
printf("\n int: %d",sizeof(int));
printf("\n long: %d",sizeof(long));
printf("\n float: %d",sizeof(float));
printf("\n double: %d",sizeof(double));
printf("\n (char *): %d",sizeof(char *));

printf("\n\nG4 Konvertierungen G0");
iret = atoi(" -123 ");
printf("\n Ergebnis atoi: %d",iret);
iret = atol(" +123123 ");
printf("\n Ergebnis atol: %ld",iret);
fret = atof(" -123.30e7 ");
printf("\n Ergebnis atof: %f",fret);
iret = isascii('A');
printf("\n Ergebnis isascii fuer A: %d",iret);
iret = isalpha(' ');
printf("\n Ergebnis isalpha fuer : %d",iret);

printf("\n\n G4 float und double Test in printf() G0");
putchar(BELL);
printf("\n 4.5 als float %f",4.5);
printf("\n 4.5 scientific float %e",4.5);
printf("\n 4.5 in g Format %g",4.5);

weiter();
printf("\n\n G4 mathematische Routinen G0");
putchar(BELL);
pi = 3.1415926;
dv = sin(45.00 * pi / 180.00);

```

```

dw = sin(89.90 * pi / 180.00);
printf("\n pi = %g",pi);
printf("\n sin(45) = %g          sin(89,9) = %g",dv,dw);
dv = cos(45.00 * pi / 180.00);
dw = sin(89.90 * pi / 180.00);
printf("\n cos(45) = %g          cos(89,9) = %g",dv,dw);
dv = tan(00.10 * pi / 180.00);
dw = tan(89.90 * pi / 180.00);
dx = dw;
printf("\n tan(00,1) = %g          tan(89,9) = %g",dv,dw);
dv = 1/(tan(00.10 * pi / 180.00));
dw = 1/(tan(89.90 * pi / 180.00));
printf("\n cot(00,1) = %g          cot(89,9) = %g",dw,dv);
dv = ((dw * 100) / dx);
printf("\n Abweichung = %g Prozent ",dv);
dv = tan(45.00 * pi / 180.00);
dw = 1/(tan(45.00 * pi / 180.00));
printf("\n tan(45) = %g          cot(45) = %g",dv,dw);
dv = log(15.1542623);
dw = log(dv);
printf("\n\n ln(15.5) = %g          ln(ln15) = %g",dv,dw);
dx = (dw * 100) - 100;
printf("\n Abweichung = %g Prozent ",dx);
dv = log10(10.0);
printf("\n log10(10.0) = %g",dv);
dv = sqrt(2.0);
printf("\n sqrt(2.0) = %g",dv);
dv = sqrt(sqrt(122.2222));
dw = sqrt(sqrt(dv));
dv = sqrt(sqrt(dw));
dw = sqrt(sqrt(dv));
dv = ((dw * 100) / 1.01895) - 100;
printf("\n\n");
printf(" 8te Wurzel aus 122.2222 = %g Abweichung = %g Prozent",dw,dv);
printf("\n\n");

weiter();
printf("\n\n G4 Test Basis-IO G0\n");
/* mit Test, ob write per default transparent ist */
printf("\n GA Test Basis-IO G0");
printf("\n File anlegen ");
fd1 = creat("xx.test",0x01c0); /* identisch chmod 0700 */
printf("\n 36 Zeichen reinschreiben ");
iret = write(fd1,"a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,r,s\n",36);
printf("\n File schlieszen ");
close(fd1);
printf("\n und wieder oeffnen ");
fd1 = open("xx.test",0);
printf("\n\n GA Ergebnis open = G0 %d",fd1);
printf("\n Das geschriebene wieder lesen ");
iret = read(fd1,puffer,36);
printf("\n Ausgabe auf Terminal mit Basis-IO \n ");
putchar(BELL);
write(1,"\n GA Ergebnis read = G0 ",27);
write(1,puffer,36);
printf("\n pruefen, ob \n zu LF+CR expandiert wurde ");
iret = read(fd1,puffer,1);
if (iret <= 0)
    printf("\n\n\t\t write-read arbeiten transparent");
else printf("\n\n\t\t G6 write-read arbeiten nicht transparent");
putchar(BELL);
close(fd1);
printf("\n\n");

weiter();

```

4

```
printf("\n\n G4 Stringroutinen G0");
printf("\n  strcmp  : %d", strcmp("ABC", "abc"));
printf("\n  strncmp : %d", strncmp("ABC", "abcdef", 4));
printf("\n  strcpy  : %s", strcpy(puffer, "vwxyz"));
printf("\n  strncpy : %s", strncpy(puffer, "abcdefg", 3));
```

```
cadr = strchr("ABCDEFGH", 'D');
printf("\n  strchr  : %lx", cadr);
cadr = strchr("ABCDEFGH", 'X');
printf("\n  strchr  : %lx", cadr);
```

```
weiter();
printf("\n\n G4 Bildschirmfuellung Standard printf G0");
```

```
/* Zeit: 33 Sekunden fuer 200 Zeilen */
```

```
ypos = 0;
```

```
printf("\n\n");
```

```
for (n=0; n < 100; n++)
```

```
{
  if (++ypos >= 25) ypos = 1;
```

```
  showline(ypos,
```

```
  "abcdefghijklmnopqrstuvwxy  ABCDEFGHIJKLMNOPQRSTUVWXYZ  0123456789");
```

```
  }
  printf("\n\n\n\n");
```

```
weiter();
```

```
putchar(BELL);
```

```
printf("\n\n\n\n\t\t\t\t\tG4  E N D E \n");
```

```
printf("\n\t\t\t\t\tG4  E N D E \n");
```

```
printf("\n\t\t\t\t\tG4  E N D E \n");
```

```
printf("\n\tG4  E N D E \n");
```

```
printf("\nG4  E N D E \n\n\n\n");
```

```
putchar(BELL);
```

```
3
```