

```

#include <funktion.h>
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <prtscr.h>

int _xwin, _ywin, g1=DETECT, g2, maxfun, p=0, li=1;
double xmin, xmax, ymin, ymax, plusx, plusy, qx, qy, add=0.01;

void *retbuf;

unsigned _stklen=0xFFFFL;
struct f
{
    char f[160];
    int col;
} funk[10];

void openwindow(int x, int y, int x1, int y1, int rcolor, int clear)
{
    int old;

    retbuf=malloc(imagesize(x, y, x1, y1));
    if (retbuf==NULL) exit(3);
    getimage(x, y, x1, y1, retbuf);
    setviewport(x, y, x1, y1, 1);
    if (clear) clearviewport();
    old=getcolor();
    setcolor(rcolor);
    line(0, 0, x1-x, 0);
    line(x1-x, 0, x1-x, y1-y);
    line(x1-x, y1-y, 0, y1-y);
    line(0, y1-y, 0, 0);
    _xwin=x; _ywin=y;
    setcolor(old);
}

void closewindow()
{
    setviewport(0, 0, getmaxx(), getmaxy(), 1);
    putimage(_xwin, _ywin, retbuf, COPY_PUT);
    free(retbuf);
}

int prealx(double x)
{
    return((int)(qx*(x-xmin)));
}

int prealy(double y)
{
    return(getmaxy()-(int)(qy*(y-ymin)));
}

void achsen()
{
    double i, x, z;
    char nr[10];

    setlinestyle(DOTTED_LINE, 0, 1);
    if (!p)
        for (i=xmin; i<=xmax; i+=plusx)
            if (!i) continue;
            else
            {
                setcolor(WHITE);
                if (floor(i)==i) sprintf(nr, "%+d", (int)i);
                else
                    sprintf(nr, "%+.1f", i);
                if (i!=xmin && i!=xmax) outtextxy(prealx(i)-(strlen(nr)*4), prealy(ymax)+2, nr);
                setcolor(LIGHTGREEN);
                line(prealx(i), prealy(ymin), prealx(i), prealy(ymax));
            }
}

```

```

    }
else
{
    double hpx;

    hpx=plusx/M_PI;
    for (z=floor(xmin/M_PI); z*M_PI<=xmax; z+=hpx)
    {
        x=z*M_PI;
        if (!z) continue;
        else
        {
            setcolor(WHITE);
            if (floor(z)==z)
                if (z== 1)
                    sprintf(nr,"π");
                else
                if (z== -1)
                    sprintf(nr,"-π");
                else
                    sprintf(nr,"%+dn", (int) z);
            else
                sprintf(nr,"%+.1fπ", z);
            if (x!=xmin && x!=xmax) outtextxy(prealx(x)-(strlen(nr)*4),prealy(ymax)+2,nr);
            setcolor(LIGHTGREEN);
            line(prealx(x),prealy(ymin),prealx(x),prealy(ymax));
        }
    }
}
for (i=ymin; i<=ymax; i+=plusy)
    if (!i) continue;
    else
    {
        setcolor(WHITE);
        if (floor(i)==i) sprintf(nr,"%+d", (int) i);
        else
            sprintf(nr,"%+.1f", i);
        if (i!=ymin && i!=ymax) outtextxy(prealx(xmin),prealy(i)-4,nr);
        setcolor(LIGHTGREEN);
        line(prealx(xmin),prealy(i),prealx(xmax),prealy(i));
    }
setlinestyle(SOLID_LINE,0,1);
setcolor(LIGHTGRAY);
line(prealx(0),prealy(ymin),prealx(0),prealy(ymax));
line(prealx(xmin),prealy(0),prealx(xmax),prealy(0));
line(prealx(xmax),prealy(0),prealx(xmax)-4,prealy(0)-4);
line(prealx(xmax),prealy(0),prealx(xmax)-4,prealy(0)+4);
line(prealx(0),prealy(ymax),prealx(0)-4,prealy(ymax)+4);
line(prealx(0),prealy(ymax),prealx(0)+4,prealy(ymax)+4);
}

int plot(struct f *funk)
{
    double x,y;
    int mx,my;
    char s[160];

    _clear87();
    _fpreset();
    strcpy(s,funk->f);
    setmul(s);
    delspac(s);
    x=_X=xmin;
    if (li)
    {
        mx=prealx(xmin); my=prealy(f(s)); x=xmin+add;
        setcolor(funk->col);
    }
    do
    {
        y=f(s);
        if (li) line(mx=prealx(x),my=prealy(y),mx,my);
        else

```

```

        putpixel(prealx(x),prealy(y),funk->col);
        x+=add;
        _X=x;
    }
    while (x<=xmax && !bioskey(1));
    if (bioskey(1)) { bioskey(0); return 1; }
    return(0);
}

void newfunk()
{
    int z,i;
    char s[160],y[160],x[160];

    closegraph();
    printf("Anzahl Funktionen : ");
    gets(x); i=atoi(x);
    for (z=maxfun; z<maxfun+i; z++)
    {
        printf("\n%2d) Funktion f(x)=",z+1); gets(s);
        strcpy(funk[z].f,strupr(s));
        printf("    Farbe der Funktion : "); gets(y);
        funk[z].col=atoi(y);
    }
    maxfun=maxfun+i;
    initgraph(&g1,&g2,"");
}

void newachs()
{
    char x[160],y[160],s[160],*m;

    closegraph();
    printf("\nFunktionsgrenzen xmin,xmax (%f,%f): ",xmin,xmax); gets(x);
    printf("Funktionsgrenzen ymin,ymax (%f,%f): ",ymin,ymax); gets(y);
    if (*x!=0)
    {
        *(m=strchr(x,', '))=0;
        strcpy(s,x);    xmin=f(s);
        strcpy(s,m+1); xmax=f(s);
    }
    if (*y!=0)
    {
        *(m=strchr(y,', '))=0;
        strcpy(s,y);    ymin=f(s);
        strcpy(s,m+1); ymax=f(s);
    }

    printf("\nBei n-notation ist nur das x-Intervall als x*n anzugeben\n");
    printf("Bitte geben Sie das x-Intervall an (%f): ",plusx); gets(x);
    printf("Bitte geben Sie das y-Intervall an (%f): ",plusy); gets(y);
    if (*x!=0) plusx=f(x);
    if (*y!=0) plusy=f(y);

    printf("\nBitte geben sie nun die additive Konstante ein (%f): ",add);
    gets(x); if (*x!=0) add=f(x);

    if (li) strcpy(x,"Ja"); else strcpy(x,"Nein");
    printf("\nSollen die einzelnen Punkte mit Linien verbunden werden? (%s): ",x);
    gets(y); if (*y!=0) li=(toupper(*y)=='N' ? 0:1);

    initgraph(&g1,&g2,"");

    qx=(double)(getmaxx())/fabs(xmax-xmin);

    qy=(double)(getmaxy())/fabs(ymax-ymin);
}

void funks()
{
    int z,i;
    char s[160],x[160],y[160];

```

```

closegraph();
printf("Anzahl Funktionen : ");
gets(x); i=atoi(x);
for (z=0; z<i; z++)
{
    printf("\n%2d) Funktion f(x)=",z+1); gets(s);
    strcpy(funk[z].f,strupr(s));
    printf("    Farbe der Funktion : "); gets(y);
    funk[z].col=atoi(y);
}
maxfun=i;
newachs();
}

void draw()
{
    int z,ab;
    do
    {
        char str[40];

        sprintf(str,"3) Achsennotation (%s)",p ? "π":"Zahlen");
        openwindow((getmaxx()/2)-(34*4),(getmaxy()/2)-42,
            (getmaxx()/2)+(34*4),(getmaxy()/2)+41,YELLOW,1);
        setcolor(LIGHTCYAN);
        outtextxy(4, 3,"Wahl : ");
        outtextxy(4,15,"1) Eingabe von neuen Funktionen");
        outtextxy(4,24,"2) Eingabe einer zusätz. Funktion");
        outtextxy(4,33,str);
        outtextxy(4,42,"4) Neue Achseneinteilung");
        outtextxy(4,51,"5) Funktionen zeichnen");
        outtextxy(4,60,"6) Drucken");
        outtextxy(4,73,"7) QUIT");
        z=getch();
        closewindow();
        ab=0;
        switch (z)
        {
            case '1': funks();
                achsen();
                for (z=0; (z<maxfun && !ab); z++) ab=plot(&funk[z]);
                if (!ab) getch();
                break;

            case '2': newfunk();
                achsen();
                for (z=0; (z<maxfun && !ab); z++) ab=plot(&funk[z]);
                if (!ab) getch();
                break;

            case '3': p=!p;
                if (!maxfun) break;
                clearviewport();
                achsen();
                for (z=0; (z<maxfun && !ab); z++) ab=plot(&funk[z]);
                if (!ab) getch();
                break;

            case '4': newachs();
            case '5': if (!maxfun) break;
                clearviewport();
                achsen();
                for (z=0; (z<maxfun && !ab); z++) ab=plot(&funk[z]);
                if (!ab) getch(); break;
            case '6': if (!maxfun) break;
                else
                    printscr(); break;

        }
    }
    while (z!='7');
}

void main()

```

```

{
    xmax=ymax=10;
    xmin=ymin=-10;
    plusx=plusy=1;
    RAD=0;
    clrscr();
    registerbgidriver(EGAVGA_driver);
    registerbgifont(small_font);
    registerbgifont(sansserif_font);
    registerbgifont(triplex_font);
    registerbgifont(gothic_font);

    printf ("Die Zahl  $\pi$  (PI) wird als chr(227) oder mit dem Zeichen '~' angegeben!\n");
    printf ("(C) by Daniel Becker in 1992\n");
    printf ("Taste -->\n");
    getch();
    _E=exp((double)(1));
    initgraph(&g1,&g2,"");
    draw();
    closegraph();
}SUB

```