

```

/*****
/*
/*   Baum aus der Postfixform
/*   Verwendet STACK.C vom 24.04.99
/*
/*   (C) by Daniel Becker
/*                               Letzte Änderung: 24.04.99
/*
*****/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <math.h>

```

```

/* Vorinitialisiert auf NULL, damit kein Aerger passieren kann... */
/* der Typ des Pointers ist "unsigned char", da ich nur dann wirklich */
/* eine Byteweise addition der Adressen erwarten kann. */

```

```

unsigned char *stack=NULL,*stk_pointer,*s_ende;
int s_element_size;

```

```

void init_stack(int elements,int size)
{
    if (stack!=NULL) return; /* Stack ist bereits initialisiert! */

    stack=(unsigned char *)malloc(elements*size);
    s_element_size=size;
    stk_pointer=stack;
    s_ende=stack+(elements*size);
}

```

```

void clear_stack()
{
    if (stack!=NULL) free(stack); /* Nur initialisierte Stacks freigeben */
    stack=stk_pointer=s_ende=NULL; /* stk_empty funktioniert auch jetzt */
}

```

```

int stk_empty()
{
    return(stk_pointer==stack);
}

```

```

int stk_full()
{
    return(stk_pointer==s_ende);
}

```

```

void push(unsigned char *c)
{
    if (stk_full())
        puts("Stack overflow");
    else
    {
        memcpy(stk_pointer,c,s_element_size); /* MemCopy(dest, Source, len); */
        stk_pointer+=s_element_size;
    }
}

```

```

unsigned char *pop()
{
    if (stk_empty())
        puts("Stack underflow");
    else
    {
        stk_pointer-=s_element_size;
        return(stk_pointer);
    }
}

```

```

void smart_pop(unsigned char *c)
{
    if (stk_empty())
        puts("Stack underflow");
}

```

```

else
{
    stk_pointer-=s_element_size;
    memcpy(c,stk_pointer,s_element_size);
}
}

unsigned char *top_of_stack()
{
    return(stk_pointer-s_element_size);
}
/***** STACK ENDE *****/

unsigned char translat[]=";=+-*/^() ";
/* aus UPN.C */

struct node
{
    char key;
    struct node *links, *rechts;
} *wurzel;

void eleminiere_baum(struct node *b)
{
    if (b->rechts) eleminiere_baum(b->rechts);
    if (b->links) eleminiere_baum(b->links);
    free(b);
}

void print_baum(struct node *n, int rek_depth, int pos,int to)
{
    int x,y;

    y=rek_depth+1;
    x=pos+(40/pow(2,rek_depth)*to);

    if (n->rechts) print_baum(n->rechts, rek_depth+1, x, +1);
    if (n->links) print_baum(n->links, rek_depth+1, x, -1);

    gotoxy(x,y);
    printf("%c",n->key);
}

void main()
{
    char postfix[256],*p;
    struct node *n;

    /* Zeiger sind 4 Byte lang... */
    init_stack(256,sizeof(struct node *));

    printf("Bitte geben Sie die Postfixform ein (mit Semikolon!):\n");
    gets(postfix);
    for (p=postfix; *p && *p!=';'; p++)
    {
        n=(struct node *)malloc(sizeof(struct node));
        n->key=*p; n->links=NULL; n->rechts=NULL;
        if (strchr(translat,*p)) /* Operator? */
        {
            smart_pop((unsigned char *)&(n->rechts));
            smart_pop((unsigned char *)&(n->links));
        }
        push((unsigned char *)&n);
    }
    wurzel=(struct node *) (* (unsigned long *) (pop()));
    /* Sollte das letzte Element sein */
    clrscr();
    print_baum(wurzel,0,0,1);
    bioskey(0);
    clrscr();
    eleminiere_baum(wurzel); /* Wech damit..... */
}SUB

```