

```
/*
 *
 * Testparser fuer eine kleine Grammatik
 * (C) by Daniel Becker und Ralf Schwaecke
 * Letzte Aenderung 6.5.99
 *
 */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

/* Grammatik:
   S= x|(A)
   A= SB
   B= {+S}
 */

char ch,Wort[80],str[80],*w;
int tested=0;

char FS[]="x(", FB[]="+"; /* Die First's, da LL(1)*/

void A();
void B();
void S();

char getnextchar()
{
    return(*(w++));
}

int first(char c, char *first_str) /* "c in first_str" */
{
    return(strchr(first_str,c)!=NULL);
}

void error(char *s)
{
    printf("\n%s\n\n",s);
    exit(3);
}

void A()
{
    S();
    B();
}

void B()
{
    while (first(ch,FB)) /* Zeichen ch muss FIRST von B sein */
    {
        ch=getnextchar(); /* Ist immer ein '+' */
        S();
    }
}

void S()
{
    if (first(ch,FS))
    {
        Wort[tested++]=ch;
        switch (ch)
        {
```

```
    case 'x' : Wort[tested++]=ch=getnextchar();
              break;
    case '(' : ch=getnextchar();
              A();
              /* Jetzt MUSS eine Klammer zu kommen !*/
              if (ch==')')
                  Wort[tested++]=ch=getnextchar();
              else
                  error("Bracked closed expced!");
              break;
    }
}
else error("Kein First von S");
}

void main()
{
    printf("Bitte zu testendes Wort eingeben: "); scanf("%s",&str);
    w=str;
    ch=getnextchar();
    S();
    if (ch!='$')
        error("Unexpected End of Source.");
    Wort[tested]=0; /* Null-Terminierung von String in C */
    puts(Wort);
}SUB
```