

```

1  program UPN;
2  uses Crt;
3  label beginn, weiter, m1, m2, ende;
4
5  var erg,ad : integer;
6
7  begin
8    clrscr;
9    write('Bitte arithmetischen Ausdruck in UPN eingeben : ');
10   asm
11   begin:
12       mov ah,1
13       int 21h           { Zeichen in al einlesen und ausgeben }
14       mov ah,0
15       cmp al,48        { Eingabe < 48 ? }
16       jb weiter       { wenn ja, gehe zu weiter }
17       cmp al,61       { Gleichheitszeichen ? }
18       je ende         { Wenn ja, beende das Programm }
19       sub al,48       { 48 abziehen, weil Eingabe Zahl ist }
20       mov ah,0       { High-Bits auf 0 setzen }
21       push ax        { eingegebene Zahl auf den Stack }
22   weiter:
23       cmp al,43       { Ist das Zeichen ein + ? }
24       jne m1         { Wenn nicht, gehe zu m1 }
25       pop ax         { Hole Summand 1 vom Stack }
26       pop cx         { Hole Summand 2 vom Stack }
27       add ax,cx      { Addiere }
28       push ax        { Zwischenergebnis auf den Stack }
29   m1:
30       cmp al,42       { Ist das Zeichen ein * ? }
31       jne m2         { Wenn nicht, gehe zu m2 }
32       pop ax         { Hole Faktor 1 von Stack }
33       pop cx         { Hole Faktor 2 von Stack }
34       mul cx         { Multipliziere und speichere in ax }
35       push ax        { Zwischenergebnis auf den Stack }
36   m2:
37       jmp beginn     { Springe zum Anfang }
38   ende:  pop ax      { Endergebnis vom Stack holen }
39         mov erg,ax   { Ergebnis in erg schreiben }
40   end;
41   writeln;
42   write('Ergebnis : ',erg); readln; { Ausgabe }
43   end.

```