ALGEBRA BY COMPUTER.

W.W.Sawyer.

Sometimes, when working on a complicated piece of algebra we wish a computer could relieve us of the mechanical parts of the work. It would be quite difficult to make a program that would deal with all the questions in a traditional textbook on elementary algebra. However, if we confine ourselves to a single type of problem, things are simpler.

Recently I was making an investigation which involved expressions such as

$$\frac{[(x+1)(x+2)(2x+1)(2x+3)]^2}{4(4x+1)(4x+3)^2(4x+9)}$$

It was necessary to carry out the multiplications and divisions involved. I had never previously tried to make a program for algebra and I imagined it would be quite hard. Surprisingly it turned out to be quite straightforward.

Description of the program.

The program in B.B.C. Basic, given at the end of this article,is able to carry out calculations involving addition, subtraction, multiplication or division of polynomials in one variable.The two polynomials being dealt with at any stage are specified by the arrays A( ) and B( ); the result of the operation is shown in an array C( ). An array R( ) is provided to show the remainder when division is being done. There are also arrays S( ) and T( ) in which results can be stored for later use.

The array A( ) specifies a polynomial a(x) as

$$a(x)=A(0)+A(1)x+A(2)x^2+\dots A(29)x^{29}$$

Polynomials b(x) and c(x) are characterised similarly by the numbers in B( ) and C( ). Many of these numbers of course will be nought. To save printing a lot of noughts, a number indicating the highest power of x that occurs in a(x) is placed in A(30) and in the same way the degrees of b(x) and c(x) are shown in B(30) and C(30).

After finding the result, c(x), we may want to apply some further operation to it, so we need to copy the figures in C( ) into A( ). We call this operation "C to A". To carry it out, all we need to do is to choose option 1 in line 140. In that line option 2, "New A" means that we are to feed fresh data into A( ). If we choose option 2, the computer will guide us through the necessary steps. These are specified in the procedure "Load A" in lines 1000 to 1110.

Lines 140 and 150 thus allow us to fill A( ) and B( ) either with new inputs or with results already stored in C( ),S( ) and T( ).

The stores S( ) and T( ) are necessary if we want to make future use of some result. If the result were left only in C( ) it would be over-written and lost when the next operation was performed. Accordingly line 100 gives us the chance to tuck the contents of C( ) safely away in S( ) or T( ), before we proceed to the next operation.

Once the desired polynomials are in A( ) and B( ), we have to indicate which operation is to be performed. Lines 300 to 340 deal with this. They summon the procedures for +,

-, * and /.

Naturally the procedures for adding (lines 10000 to 10180) and subtracting (11000 to 11180) are quite simple. It is perhaps surprising that multiplication, (12000 to 12270), and division, (13000 to 13330) can be carried out by exactly the same steps as we would use in paper-and-pencil calculations.

In dividing it is understood that a(x) is to be divided by b(x). If a(x) and b(x) are to be polynomnials calculated in advance, a(x) should be stored in S( ) and b(x) in T( ). Options 3 in lines 140 and 190 will copy these into A( ) and B( ).

When any operation is performed, the procedure automatically prints out an equation showing the polynomials a(x) and b(x), the nature of the operation performed, and the result c(x). We thus have a permanent record of the calculations made, and it is easy to see if we have typed in any wrong numbers.

A record is also printed when a result is stored in S( ) or T( ). This is helpful when we need to use that result later.

The present program does not provide for the storage transfers "C to B" and "T to A". it would be quite easy to include these. The lists of options would then be a little longer, but the programming would be quite simple,as may be seen by examining the procedure "T to B" in lines 18000 to 18060. With the help of the function keys, such procedures can be written with very little labour. It is also easy to provide further storage places if desired, or to make provision for polynomials of degree higher than 29.

Structure.

The program involves PROCs, which are easily checked separately if some error should be typed into the program. There is one GOTO in line 350, to give a start on the next operation, after an operation has been completed. This does not seem objectionable. If a mistake is made in the course of inputting A( ) and B( ), but before the operation has been implemented, it can be corrected by "Escape. GOTO 100". No doubt a more elegant way of achieving these two aims can be found.

A Sample Calculation and the Program.

In conclusion, we have the record produced when the complicated algebraic fraction, given at the beginning of this article, was handled by the program, and the program itself. Line 0, "ALG3" is simply a label, indicating that this program was arrived at after two modifications of a first attempt at an algebra program.

```
1 1 *
2 1 =
2 3 1
----------
2 3 1 *
1 2 =
2 7 7 2
----------
2 7 7 2 *
3 2 =
6 25 35 20 4
----------
C TO S
S TO B
6 25 35 20 4 *
6 25 35 20 4 =
36 300 1045 1990 2273 1600 680 160 16
----------
C TO S
4 *
1 4 =
4 16
----------
4 16 *
3 4 =
12 64 64
----------
12 64 64 *
3 4 =
36 240 448 256
----------
```

```
36 240 448 256 *
9 4 =
324 2304 4992 4096 1024
----------
C TO T
S TO A
T TO B
36 300 1045 1990 2273 1600 680 160 16 /
324 2304 4992 4096 1024 =
9.1003418E-2 0.21875 0.212890625 9.375E-2 1.5625E-2
REMAINDER
6.51489258 19.453125 17.734375 4.375
-----------
```

```
 0 PRINT"ALG3"
 1 PRINT"POLYNOMIALS,ADD, SUBTRACT"
 2 PRINT"MULTIPLY OR DIVIDE."
10DIMA(30)
20DIMB(30)
30DIMC(30)
40DIMR(30)
50 DIM S(30)
60 DIM T(30)
100PRINT"1. C TO S 2. C TO T 3.NEITHER"
110INPUT Q
120IF Q=1 THEN PROCCTOS
130IF Q=2 THEN PROCCTOT
140PRINT"1. C TO A 2. NEW A 3.S TO A"
150INPUT Q
160IF Q=1 THEN PROCCTOA
170IF Q=2 THEN PROCLOADA
180IF Q=3 THEN PROCSTOA
190PRINT"1. NEW B 2. S TO B 3.T TO B"
200INPUT Q
210IF Q=1 THEN PROCLOADB
220IF Q=2 THEN PROCSTOB
230IF Q=3 THEN PROCTTOB
```

```
300PRINT"+,-,* OR / ?": INPUT Q$
310IF Q$="+" THEN PROCADD
320IF Q$="-" THEN PROCMIN
330IF Q$="*" THEN PROCMULT
340IF Q$="/" THEN PROCDIV
350 GOTO 100
999 END
1000DEFPROCLOADA
1010PRINT"DEGREE OF A";:INPUT A(30)
1020I=0
1030REPEAT
1040PRINT"A(";I;")=";:INPUT A(I)
1050I=I+1
1060UNTIL I)A(30)
1070REPEAT
1080A(I)=0
1090I=I+1
1100UNTIL I)29
```

```
1110ENDPROC
2000DEFPROCCTOA
2020I=0
2030REPEAT
2040A(I)=C(I)
2050I=I+1
2060UNTIL I)30
2070 ENDPROC
3000DEFPROCLOADB
3010 PRINT"DEGREE OF B";:INPUT B(30)
3020I=0
3030REPEAT
3040PRINT"B(";I;")=";: INPUT B(I)
3050I=I+1
3060UNTIL I)B(30)
3[illegible]REPEAT
3070REPEAT
```

```
3080B(I)=0
3090I=I+1
3100UNTIL I)29
3110ENDPROC
10000DEFPROCADD
10010IF A(30))=B(30) THEN C(30)=A(30) ELSE C(30)=B(30)
10020FOR I=0 TO29
10030C(I)=A(I)+B(I)
10040NEXT
10045 VDU2
10050FOR I=0 TO A(30)
10060PRINT;A(I);" ";
10070NEXT
10080PRINT"+"
10090FOR I=0 TO B(30)
10100PRINT;B(I);" ";
10110NEXT
```

```
10120PRINT"="
10130FOR I=0 TO C(30)
10140PRINT;C(I);" ";
10150NEXT
10160PRINT" "
10170PRINT"-----------"
10175 VDU3
10180 ENDPROC
11000DEFPROCMIN
11010IF A(30))=B(30) THEN C(30)=A(30) ELSE C(30)=B(30)
11020FOR I=0 TO 29
11030C(I)=A(I)-B(I)
11040NEXT
11045 VDU2
11050FOR I=0 TO A(30)
11060PRINT;A(I);" ";
11070NEXT
```

```
11080PRINT"-"
11090FOR I=0 TO B(30)
11100PRINT;B(I);" ";
11110NEXT
11120PRINT"="
11130FOR I=0 TO C(30)
11140PRINT;C(I);" ";
11150NEXT
11160PRINT" "
11170PRINT"------------"
11175 VDU3
11180  ENDPROC
12000DEFPROCMULT
12010C(30)=A(30)+B(30)
12020FOR I=0 TO 29
12030C(I)=0
12040NEXT
12050J=0
```

```
12060REPEAT
12070K=0
12080REPEAT
12090C(J+K)=C(J+K)+A(J)*B(K)
12100K=K+1
12110UNTIL K)B(30)
12120J=J+1
12125 VDU2
12130UNTIL J)A(30)
12140FOR I=0 TO A(30)
12150PRINT;A(I);" ";
12160NEXT
12170PRINT"*"
12180FOR I=0 TO B(30)
12190PRINT;B(I);" ";
12200NEXT
12210PRINT"="
12220FOR I=0 TO C(30)
```

```
12230 PRINT;C(I);" ";
12240NEXT
12250PRINT" "
12260PRINT"----------"
12265 VDU3
12270ENDPROC
13000DEFPROCDIV
13010IF B(30))A(30) THEN ENDPROC
13020C(30)=A(30)-B(30)
13030FOR I=0 TO 29
13040R(I)=A(I)
13050NEXT
13060P=A(30):Q=B(30)
13070REPEAT
13080M=R(P)/B(Q)
13090C(P-Q)=M
13100FOR J=0 TO Q
13110R(P-J)=R(P-J)-M*B(Q-J)
```

```
13120NEXT
13130P=P-1
13140UNTIL P(Q
13145 VDU2
13150FOR I=0 TO A(30)
13160PRINT;A(I);" ";
13170NEXT
13180PRINT"/"
13190FOR I=0 TO B(30)
13200PRINT;B(I);" ";
13210NEXT
13220PRINT"="
13230FOR I=0 TO C(30)
13240PRINT;C(I);" ";
13250NEXT
13260PRINT" "
13270PRINT"REMAINDER"
13280FOR I=0 TO P
```

```
13290PRINT;R(I);" ";
13300NEXT
13310PRINT" "
13320PRINT"-----------"
13325 VDU3
13330ENDPROC
14000DEFPROCCTOS
14010FOR I=0 TO 30
14020S(I)=C(I)
14030NEXT
14040VDU2
14050PRINT"C TO S"
14060VDU3
14070ENDPROC
15000DEFPROCCTOT
15010FOR I=0 TO 30
15020T(I)=C(I)
15030NEXT
```

```
15040VDU2
15050PRINT"C TO T"
15060VDU3
15070ENDPROC
16000DEFPROCSTOA
16010FOR I=0 TO 30
16020A(I)=S(I)
16030NEXT
16040VDU2
16050PRINT"S TO A"
16055 VDU3
16060ENDPROC
17000DEFPROCSTOB
17010FOR I=0 TO 30
17020B(I)=S(I)
17030NEXT
17040VDU2
17050PRINT"S TO B"
17060VDU3
17070ENDPROC
18000DEFPROCTTOB
18010FOR I=0 TO 30
18020B(I)=T(I)
18030NEXT
18040VDU2
18050PRINT"T TO B"
18055 VDU3
18060ENDPROC
)
```