

tell the Optimizing Compiler to eliminate these jumps by adding /J, /JF, or /FJ at the end of the program name. Of course if you use one of the options with an F then the special floating point routines will be added. The following sequence will compile the program named ALPHA and will not use jumps around functions and subroutines. It will, however, add the necessary special floating point routines:

EXEC OPCOM.COD

Insert program disk & then

Enter program name

?ALPHA/JF

Enter output file name

?ALPHA.COD

Another option you can specify when you compile is to remove the BASIC during program execution. Do this by following the program name with a space and then a 2. In most cases 800XL and XE owners won't need this option since it only gains about 3 or 4K of RAM. The rest of the BASIC is shifted to high RAM in these computers. 400/800 owners will gain about 17K with this option. For example:

EXEC OPCOM.COD

Insert program disk & then

Enter program name

?ALPHA/F 2

Enter output file name

?ALPHA.COD

This will compile the program named ALPHA and use the special floating point routines. When ALPHA.COD is executed the BASIC will be removed. At the completion of ALPHA.COD you will need to insert a disk with the BASIC and press RETURN.

Sample Program

As an example of using the Optimizing Compiler we will modify and compile the SIEVE.BAS program on the Advan BASIC Master disk. First insert this disk into drive 1 and type

LOAD SIEVE.BAS

Next type:

57 FAST

180 FAST END

This places FAST and FAST END into the program. Now remove the Master disk, insert one of your disks, and type:

SAVE SIEVE.FST

After the save is finished remove the disk, insert the Optimizing Compiler Master disk into drive 1, and type: