Boston Computer Society
TI-99/4A User Group
One Center Plaza
Boston, MA 02108

c-tutorial

This disk contains a number of simple c programs. Programs ending in 'c' are c.code, those ending in 's' are compiled source.code, and those ending in 'o' are assembled object code.(most The of the source codes have been eliminated to save space, but can easily be compiled as practice).On the other hand,you can use the supplied s.codes to test your assembling technique, and the o.codes to immediately try "LOAD and RUN".Those few without these designations are either short library functions or longer program files ready to run in opt.5 of Editor/Assembler. (NOT FNLWRTR opt.3!)

Some of these these programs have been modified from texts to run in c99 and others are original.This is 2nd update of an earlier tutorial; this dsk uses v2.0 of Clint Pulley's c99.

One of the advantages of c is its ability to use a "library" of functions in programs by the statement, for example, "#include dsk1.roman". With two disk drives, it is most convenient to have the "library disk" in drive 2, and use statement "#include dsk2.roman". With only one drive, the library functions (from c99 and this disk) should be copied onto a blank disk, which is then used for compiling your program, because the library must be available during compiling.(change references to #include dsk1.****)

The usual procedure for writing a c program in c99:

1. Write the program,using Editor and SAVE as "-----c"
2. Compile using c99c and the library to form "-----s"
3. Assemble to form "-----o"
4. "load and run" your object code, 'csup'(from c99 dsk), plus any obj codes used (e.g. grf1);program name is "START"
5. If desired,convert to program format by "load and run":
   a. c99pfi (from c99 disk)
   b. your object code
   c. csup
   d. other needed obj codes (e.g.PRINTF)
   e. c99pff (from c99 disk)
   f. "SAVE" (on Editor/Assembler B)

Included on this disk are the library functions:

1. change
2. conv;c
3. csort
4. cpos
5. cseg
6. max
7. min
8. prf
9. rom
10. stoi
11. strcat

12. stncpy
13. strlen

c.codes(some examples):

```
prim2c  find all primes below
          1000
pmk2c   tests integer for
          primeness
modromc prints 1-100 in roman
          numbers
rk3c    changes number to roman
          using function rom
scinc   totals your change
simp2c  simple string
          manipulation
testmxc test max function
tstgic tests getint funct
add10c  does 10 additions
cseg;c  equivalent to basic SEG$
strpos;c equiv to basic POS
intdemc getint demo
arr1c   array programs
arr2c
arr3c
arr12c
stncpydmc demo of stncpy
```

Finally, there is a progressive set of 'change for a dollar' programs, showing my steps in writing it.

```
cfdc is the original program
cdc,cds,cdo uses function,quits if
              100 entered
cd3c          uses library function
dchange       program format (e/a 5)
```

d.l.mahler
36 BOULDER ROAD,
NEWTON,MA 02159