INTRODUCTION

THIS PROGRAM IS DESIGNED TO HELP YOU LEARN ABOUT THE GRAPHICS SYMBIOSIS SYSTEM (GRASS). THE INTENTION IS TO PROVIDE YOU WITH THE FUNDAMENTAL KNOWLEDGE AND SKILLS NECESSARY TO ALLOW YOU TO USE THE THE SYSTEM. IT IS NOT INTENDED TO PROVIDE YOU WITH A COMPLETE DESCRIPTION OF ALL THE POSSIBLE WAYS IN WHICH THE SYSTEM COULD BE USED. THAT WOULD BE IMPOSSIBLE BECAUSE THE USES OF THE SYSTEM ARE LIMITED ONLY BY YOUR OWN IMAGINATION.

THERE ARE THREE PARTS TO THIS PROGRAM. DEPENDING ON YOUR PRESENT KNOWLEDGE OF THE SYSTEM, YOU MAY WISH TO GO THROUGH ALL, OR ONLY SOME, OF THESE. THE FIRST PART WILL TEACH THE COMMANDS, AND ALLOW PRACTICE IN THE SKILLS, NECESSARY TO CREATE PICTURES. THE SECOND PART WILL DO THE SAME FOR PICTURE MANIPULATION. THE THIRD PART WILL BE ON "MACRO" WRITING. EACH PART WILL ASSUME KNOWLEDGE OF THE PRECEEDING PARTS.

PART 1 PICTURE CREATION

OBJECTIVES: UPON COMPLETION OF THIS PART THE USER WILL...

1. BE ABLE TO DRAW PICTURES BY COMBINING GRASS COMMANDS, VARIABLES, AND PROGRAM LOOPS.
2. BE ABLE TO STORE AND RETRIEVE PICTURES THAT HAVE BEEN CREATED.
3. BE FAMILIAR WITH THE USE OF THE FOLLOWING GRASS COMMANDS: OPEN, PUTPOI, DELPOI, CLOSE, IF, SKIP, PUTDSK, GETDSK, DO, EDIT, AND RENAME.
4. BE FAMILIAR WITH THE USE OF THE GRAPHICS SYMBIOSIS SYSTEM (TELETYPE, PROGRAM DISPLAY, AND GRAPHICS DISPLAY).
5. BE FAMILIAR WITH THE USE OF VARIABLES AND EXPRESSIONS IN CREATING PICTURES.
6. BE ABLE TO CREATE AND EDIT PICTURE DRAWING MACROS.
7. BE FAMILIAR WITH THE USE OF LOGICAL OPERATORS IN CONDITIONAL STATEMENTS.

SECTION 1.0 ..... USING THE SYSTEM

BEFORE WE BEGIN TO TALK ABOUT THE COMMANDS YOU WILL NEED TO KNOW IN ORDER TO CREATE PICTURES IN GRASS, YOU FIRST MUST LEARN HOW TO GIVE THESE COMMANDS TO THE COMPUTOR. ALL COMMUNICATION WITH GRASS IS DONE VIA THE VIDEO CONSOLE (VT05). AS YOU TYPE THE COMMANDS YOU WILL SEE THE LETTERS PRINTED ON THE SCREEN ABOVE THE KEYBOARD. ALL INPUTS ON THE KEYBOARD MUST BE ENDED WITH A CARRAGE RETURN. THIS IS DONE BY HITTING THE KEY MARKED CR. THIS LETS GRASS KNOW THAT YOU HAVE FINISHED A LINE OR A COMMAND. THUS IF YOU WERE GOING TO TYPE IN THE TWO COMMANDS "COME" AND "GO", YOU WOULD HIT THE FOLLOWING SEQUENCE OF LETTERS: C-O-M-E-CR-G-O-CR. ON THE SCREEN YOU WOULD SEE: COME
GO

IF YOU MAKE A MISTAKE AND TYPE SOME INCORRECT CHARACTERS, YOU MAY ERASE THEM BY HITTING THE KEY MARKED "RUB OUT". THIS WORKS ONLY ON THE LINE YOU ARE CURRENTLY TYPING. THEREFORE, ONCE YOU HAVE HIT THE CARRIAGE RETURN KEY, YOU WILL NOT BE ABLE, TO CHANGE THE PREVIOUS LINE.

AS YOU PROGRESS THROUGH THIS WORKBOOK YOU WILL BE TOLD AT VARIOUS TIMES TO DO AN EXERCISE. THESE ARE DONE AT THE VIDEO CONSOLE. TO CALL EXERCISE NO. 1, FOR INSTANCE, YOU WOULD GO TO THE VT05 KEYBOARD, AND TYPE "CALL EX1" FOLLOWED BY A CARRIAGE RETURN. YOU WILL THEN FOLLOW THE INSTRUCTIONS THAT YOU WILL SEE ON THE SCREEN ABOVE THE KEYBOARD.

PICTURES IN GRASS ARE DISPLAYED ON THE LARGE SCREEN (VECTOR GENERAL). THIS SCREEN IS DIVIDED BY THREE COORDINATE AXES; X, Y, AND Z. THESE AXES ARE CENTERED ON THE SCREEN AND FIXED SUCH THAT THE X AXIS IS HORIZONTAL, THE Y AXIS IS VERTICLE, AND THE Z AXIS IS PERPENDICULAR TO THE FACE OF THE SCREEN. THESE LINES ARE NOT USUALLY DISPLAYED ON THE SCREEN BUT ARE USED TO LOCATE VARIOUS POINTS IN A PICTURE. EACH AXIS IS 2000 UNITS LONG IN BOTH THE POSITIVE AND THE NEGATIVE DIRECTIONS. THEREFORE IF WE WISHED TO DESIGNATE THE LOCATION OF A POINT IN THE LOWER RIGHT HAND CORNER OF THE SCREEN WE MIGHT DESCRIBE IT BY STATING THAT ITS LOCATION IS 1500 UNITS IN THE X DIRECTION, -1000 UNITS IN THE Y DIRECTION, AND 0 UNITS IN THE Z DIRECTION. OR X=1500, Y=-1000, Z=0. (FIGURE 1)

SINCE THE Z AXIS IS PERPENDICULAR TO THE SCREEN, YOU WILL NOT BE ABLE TO SEE THE LOCATION OF THE POINT IN THAT DIRECTION UNLESS THE AXES ARE ROTATED. IF THIS IS SOMEWHAT CONFUSING, DON'T WORRY, IT SHOULD BECOME MORE CLEAR WHEN YOU ACTUALLY START USING THE SYSTEM.

IF AT ANY TIME THINGS GET OUT OF HAND, THAT IS, YOU GET FUNNY MESSAGES THAT DON'T MAKE SENSE OR THE SYSTEM STOPS OPERATING, DON'T WORRY ABOUT IT. YOU CAN RESTART ANY EXERCISE AT ANY TIME BY TYPING A CONTROL C (HOLD DOWN THE CTRL KEY AND TYPE C),THEN TYPE RESTART, THEN CALL THE EXERCISE AGAIN.

SECTION 1.1 ..... PICTURE CREATION COMMANDS

THE CHIEF METHOD OF CREATING PICTURES IN GRASS CONSISTS OF SELECTING VARIOUS POINTS ON THE DISPLAY TO BE CONNECTED BY STRAIGHT LINES (CALLED VECTORS). THESE POINTS ARE DEFINED BY THEIR X,Y, AND Z COORDINATE VALUES. IF, FOR EXAMPLE, YOU WISHED TO DRAW A LINE REPRESENTING THE POSITIVE PORTION OF THE X AXIS, THE POINTS NECESSARY TO "DEFINE" THIS LINE (VECTOR) ARE:

POINT #1 X=0, Y=0, Z=0
POINT #2 X=2000, Y=0, Z=0

A STRAIGHT LINE DRAWN BETWEEN THESE POINTS WOULD REPRESENT THE POSITIVE X AXIS. NOTICE THAT TWO POINTS WERE NECESSARY TO DEFINE THIS LINE (YOU MIGHT REMEMBER A RULE ABOUT THIS FROM GEOMETRY). GRASS ALWAYS EXPECTS YOU TO GIVE IT TWO POINTS WHEN YOU WANT IT TO DRAW A LINE.

WE NOW COME TO THE FIRST COMMAND NECESSARY TO CREATE A GRASS PICTURE. THIS IS THE OPEN COMMAND.

*

SYNTAX: OPEN LNAME

THE OPEN COMMAND TELLS GRASS TO SET ASIDE SOME MEMORY SPACE (CALLED A FILE) FOR THE PICTURE YOU ARE ABOUT TO CREATE. LNAME IS ANY NAME YOU WISH TO SUPPLY, SIX LETTERS OR LESS, FOR YOUR PICTURE (SINGLE PICTURES ARE CALLED LEAFS WHICH IS WHY THE L IS THERE). FOR EXAMPLE:

OPEN PIX

ONCE YOU HAVE NAMED YOUR PICTURE, AND GRASS HAS SET ASIDE SOME MEMORY FOR IT, YOU INPUT THE POINTS WHICH MAKE UP YOUR PICTURE. THIS IS DONE WITH THE PUTPOI COMMAND.

SYNTAX: PUTPOI X,Y,Z,K

THIS COMMAND TELLS GRASS TO PLACE A POINT IN YOUR PICTURE AT THE LOCATION SPECIFIED BY X,Y, AND Z. THE K TELLS GRASS WHETHER OR NOT TO DRAW A LINE FROM THE PREVIOUS POINT AS FOLLOWS:

K=0 DRAW THE LINE
K=1 PLACE THE POINT IN THE PICTURE BUT DO NOT DRAW A LINE

* (NOTE) ALL COMMANDS WILL BE PRESENTED IN THIS FORMAT:

SYNTAX: COMMAND

THE SYNTAX SHOWS THE GENERAL FORM OF THE COMMAND AND HOW THE COMMAND MUST BE TYPED. THE SPELLING OF THE COMMAND AND THE SPACES AND COMMAS BETWEEN PARTS OF THE COMMAND ARE VERY IMPORTANT. THE COMMAND WILL NOT WORK IF IT IS SPELLED WRONG OR IF A SPACE OR A COMMA IS OMMITTED.

TO USE THIS COMMAND TO DRAW THE POSITIVE X AXIS YOU WOULD TYPE THE FOLLOWING:

```
PUTPOI 0,0,0,1
PUTPOI 2000,0,0,0
```

NOTICE THAT TWO COMMANDS MUST BE USED (TWO POINTS DEFINE A LINE), AND THAT THE FIRST ONE PLACES A POINT BUT DOES NOT DRAW A LINE (NO PLACE TO DRAW ONE FROM) WHILE THE SECOND ONE PLACES A POINT AND DRAWS A LINE FROM THE FIRST POINT.

THE THIRD COMMAND NECESSARY TO CREATE GRASS PICTURES IS THE CLOSE COMMAND.

SYNTAX: CLOSE

THIS COMMAND TELLS GRASS THAT YOUR PICTURE IS FINISHED. IT IS ABSOLUTELY NECESSARY IF YOU WISH TO STORE A PICTURE FOR LATER USE OR IF YOU WISH TO OPEN A NEW FILE. NOTE THAT A NEW PICTURE FILE MAY BE OPENED ONCE A PREVIOUS FILE HAS BEEN CLOSED. MANY SEPARATE PICTURES MAY BE CREATED AS LONG AS YOU DO NOT TRY TO HAVE MORE THAN ONE FILE OPEN AT A TIME.

YOU SHOULD NOW BE ABLE TO WRITE A COMPLETE PICTURE CREATING PROGRAM (AT LEAST ONE TO DRAW THE POSITIVE X AXIS). AN EXAMPLE OF THIS WOULD BE:

```
OPEN AXIS              (OPEN FILE AND NAME PICTURE)
PUTPOI 0,0,0,1         (PLACE FIRST POINT)
PUTPOI 2000,0,0,0      (SECOND POINT AND LINE DRAWN)
CLOSE                  (CLOSE PICTURE FILE)
```

MORE COMPLICATED PICTURES INVOLVE REPEATED USE OF THE PUTPOI COMMAND. TO DRAW A TRIANGLE WE MIGHT HAVE THE FOLLOWING:

```
OPEN TRI                        OPEN TRI
PUTPOI 500,0,0,1                PUTPOI 0,500,0,1
PUTPOI 0,500,0,0        OR      PUTPOI 500,0,0,0
PUTPOI -500,0,0,0               PUTPOI -500,0,0,0
PUTPOI 500,0,0,0                PUTPOI 0,500,0,0
CLOSE                           CLOSE
```

NOTICE THAT 4 PUTPOI COMMANDS WERE NECESSARY TO DRAW 3 LINES (2 POINTS PER LINE REMEMBER). NOTICE ALSO THAT THE SAME PICTURE CAN BE DRAWN IN MORE THAN ONE WAY.

A FOURTH COMMAND, WHICH IS USEFUL IN PICTURE CREATION, OR MORE PROPERLY PICTURE DESTRUCTION, IS THE DELPOI COMMAND.

SYNTAX: DELPOI

THIS COMMAND WILL DELETE (ERASE) THE POINT PLACED BY THE LAST PUTPOI COMMAND. THIS IS NORMALLY DONE WHEN YOU HAVE PLACED A POINT IN THE WRONG LOCATION. THE FOLLOWING PROGRAM WAS MEANT TO DRAW THE ENTIRE X AND Y AXES:

```
OPEN AXES
PUTPOI -2000,0,0,1
PUTPOI 2000,0,0,0
PUTPOI 0,2000,0,0
DELPOI
PUTPOI 0,2000,0,1
PUTPOI 0,-2000,0,0
CLOSE
```

DO YOU SEE WHY THE DELPOI COMMAND WAS USED? THE THIRD PUTPOI COMMAND DREW A LINE THAT SHOULD NOT HAVE BEEN IN THE PICTURE (K=0). THE DELPOI COMMAND REMOVED THAT LINE FROM THE PICTURE. THE FOURTH PUTPOI COMMAND WAS THE CORRECT ONE (K=1).

THE TIME HAS NOW COME FOR SOME PRACTICE. GO TO THE VIMS KEYBOARD, TYPE "CALL EX1", AND FOLLOW THE DIRECTIONS PRINTED ON THE SCREEN.

GOOD LUCK !!!

THE COMMANDS PRESENTED IN THIS SECTION WERE:

```
OPEN LNAME
PUTPOI X,Y,Z,K
CLOSE
DELPOI
```

SECTION 1.2 ..... VARIABLES, EXPRESSIONS, OPERATORS, AND LOOPS

IF YOU HAVE COMPLETED EXERCISE #1 (IF YOU HAVN'T YOU SHOULDN'T BE HERE, SO GO AWAY), YOU PROBABLY REALIZE BY NOW THAT DRAWING PICTURES POINT BY POINT IS A RATHER TIME CONSUMING PROCESS. HOWEVER ONE OF THE THINGS A COMPUTOR CAN DO VERY WELL IS EXECUTE A SERIES OF COMMANDS OVER AND OVER AGAIN. THIS PROGRAMMING CONCEPT IS CALLED LOOPING. NOW YOU ASK, "WHY WOULD ANYONE WANT TO EXECUTE THE SAME COMMAMDS OVER AND OVER AGAIN?" WELL, IF THEY WERE EXACTLY THE SAME WE MIGHT NOT. BUT IF WE COULD CHANGE THE COMMANDS SLIGHTLY EACH TIME THEN WE MIGHT.

ONE WAY TO CHANGE A COMMAND IS THROUGH THE USE OF SOMETHING CALLED A VARIABLE. A VARIABLE IS A SYMBOL WHICH CAN TAKE ON ANY NUMERIC VALUE YOU WISH TO ASSIGN TO IT. IN GRASS THE SINGLE LETTERS A THROUGH Z ARE VARIABLES (THERE ARE OTHERS BUT WE WONT NEED THEM JUST YET). VARIABLES ARE ASSIGNED VALUES WITH ASSIGNMENT STATEMENTS SUCH AS A=5 OR P=-5. SUCH A STATEMENT, WHEN TYPED ON THE KEYBOARD, WILL TELL GRASS TO SUBSTITUTE A 3 WHEREVER IT FINDS AN "A", OR A -5 WHEREVER IT FINDS A "P". THUS IF THE FOLLOWING WERE TYPED IN:

```
A=1000
PUTPOI A,0,A,1
```

A POINT WOULD BE PLACED AT THE LOCATION X=1000, Y=0, Z=1000.

THE GENERAL FORM OF THE ASSIGNMENT STATEMENT WHICH ASSIGNS A VALUE TO A VARIABLE IS:

```
VARIABLE=EXPRESSION
```

WHERE "VARIABLE" IS ANY LEGAL VARIABLE (SINGLE LETTERS A-Z FOR NOW), AND "EXPRESSION" IS A MIXTURE OF NUMBERS, VARIABLES, AND ARITHMETIC OPERATORS, WHICH ALWAYS EVALUATES TO A SINGLE NUMBER. FOR EXAMPLE:

```
A=3+5      (A) VARIABLE,   (3+5) EXPRESSION
R=A-2+6    (R) VARIABLE,   (A-2+6) EXPRESSION
```

THE ARITHMETIC OPERATORS WHICH CAN BE USED IN GRASS ARE:

```
+ ADDITION
- SUBTRACTION
* MULTIPLICATION
/ DIVISION
```

IN ADDITION TO VARIABLES, EXPRESSIONS CAN BE USED IN THE PUTPOI COMMAND. FOR EXAMPLE:

```
A=100
PUTPOI A,A+100,0,1
```

WOULD PLACE A POINT AT X=100, Y=200, Z=0.

ONE OF THE NICE THINGS ABOUT VARIABLES IS THAT THEY CAN BE ASSIGNED DIFFERENT VALUES IN DIFFERENT PARTS OF A PROGRAM. THUS THE FOLLOWING:

```
A=100
PUTPOI A,0,0,1
A=200
PUTPOI A,0,0,0
```

WOULD DRAW A LINE FROM X=100 TO X=200. IN ADDITION A VARIABLE CAN BE MADE TO CHANGE IT'S OWN VALUE AS IN THE FOLLOWING:

```
A=100
PUTPOI A,0,0,1
A=A+100
PUTPOI A,0,0,0
```

WHICH AGAIN DRAWS A LINE FROM X=100 TO X=200. THE STATEMENT A=A+100, TAKES THE VALUE OF "A" PRIOR TO THE EXECUTION OF THE STATEMENT, ADDS 100 TO IT, AND SETS "A" EQUAL TO THE SUM. EACH TIME THE STATEMENT IS EXECUTED THE VALUE OF "A" INCREASES BY 100.

IN ORDER TO USE THIS VALUABLE TOOL WE NEED A COMMAND WHICH WILL CAUSE THE COMPUTOR TO JUMP BACK SEVERAL STATEMENTS AND EXECUTE THEM AGAIN. IN GRASS THIS IS THE SKIP COMMAND.

SYNTAX: SKIP EXPR

THE SKIP COMMAND ALLOWS US TO MAKE THE COMPUTOR JUMP AROUND WITHIN A PROGRAM. "EXPR" STANDS FOR EXPRESSION, AND MEANS THE SAME AS BEFORE, A MIXTURE OF NUMBERS, VARIABLES, AND ARITHMETIC OPERATORS WHICH EVALUATE TO A SINGLE NUMBER. "EXPR" TELLS GRASS WHERE TO JUMP. FOR EXAMPLE:

```
A=200
PUTPOI A,0,0,1
PUTPOI 0,A,0,0
A=A+200
SKIP -3
```

THE SKIP COMMAND CAUSES THE COMPUTOR TO JUMP BACK TO THE FIRST PUTPOI COMMAND. THE TWO PUTPOI COMMANDS ARE EXECUTED OVER AND OVER AGAIN, AND EACH TIME "A" IS 200 MORE THAN THE LAST TIME. THIS IS CALLED A LOOP. (WHAT PICTURE WOULD BE DRAWN?)

THIS IS ALL VERY WELL AND GOOD, BUT SINCE COMPUTERS CAN ONLY DO WHAT THEY ARE TOLD, WE MUST TELL IT WHEN TO STOP. THIS IS DONE WITH WHAT IS CALLED A CONDITIONAL STATEMENT. THE CONDITIONAL STATEMENT IN GRASS IS THE IF COMMAND.

SYNTAX: IF VAR OPR EXPR,COMMAND

SINCE THIS LOOKS LIKE A MISTAKE INSTEAD OF A COMMAND, I WILL GO THROUGH EACH PART IN TURN. "IF" IS THE COMMAND ITSELF. "VAR" IS ANY VARIABLE JUST AS BEFORE. "OPR" STANDS FOR WHAT IS CALLED A LOGICAL OPERATOR.(I'LL EXPLAIN IN A SECOND). "EXPR" IS ANY VALID EXPRESSION JUST AS BEFORE. "COMMAND" IS ANY LEGAL GRASS COMMAND SUCH AS PUTPOI OR SKIP.

NOW ABOUT THOSE LOGICAL OPERATORS. IN GRASS THEY ARE THE FOLLOWING:

LT (LESS THAN)	LE (LESS THAN OR EQUAL)
GT (GREATER THAN)	GE (GREATER THAN OR EQUAL)
EQ (EQUAL)	NE (NOT EQUAL)

THESE OPERATORS MEAN JUST WHAT THEY SAY. WHEN THEY ARE PUT IN AN IF COMMAND THEY TEST THE CONDITION OF THE VARIABLE. THAT IS WHY IT IS CALLED A CONDITIONAL STATEMENT. THE STATEMENT (IF B LT 100,SKIP -2) TESTS TO SEE IF B IS LESS THAN 100. IF THE ANSWER IS TRUE THEN THE PORTION OF THE STATEMENT AFTER THE COMMA IS EXECUTED. IF THE ANSWER IS NOT TRUE THEN THE LINE FOLLOWING THE IF COMMAND IS EXECUTED. HERE IS AN EXAMPLE:

```
B=0
B=B+100
IF B LE 1000,SKIP -1
PUTPOI B,0,0,1
```

IN THIS EXAMPLE B STARTS OUT EQUAL TO 0. WHEN THE SECOND STATEMENT IS EXECUTED B IS EQUAL TO 100. THE THIRD STATEMENT TESTS TO SEE IF B IS LESS THAN OR EQUAL TO 1000. IF IT IS, THEN THE SECOND STATEMENT IS EXECUTED AGAIN AND B NOW EQUALS 200. THIS CONTINUES UNTIL B EQUALS 1100 AT WHICH TIME B IS NO LONGER LESS THAN OR EQUAL TO 1000, AND THE FOURTH STATEMENT, PUTPOI, IS EXECUTED. THIS PROGRAM PLACES A POINT AT X=1100, Y=0, Z=0.

HERE IS ANOTHER EXAMPLE:

```
OPEN PIX
A=0
PUTPOI A,0,0,1
PUTPOI A,500,0,0
A=A+50
IF A GT 500,SKIP 2
SKIP -4
CLOSE
```

THIS IS AN ENTIRE PICTURE DRAWING PROGRAM. IT WORKS LIKE THIS. FIRST THE PICTURE IS OPENED, AND THE VARIABLE IS SET TO 0. THEN TWO POINTS ARE PLACED AND A LINE IS DRAWN. THEN THE VALUE OF THE VARIABLE IS INCREASED BY 50 AND THIS NEW VALUE IS TESTED. IF THE VARIABLE IS GREATER THAN 500 THE PROGRAM CLOSES THE PICTURE, IF NOT THE PROGRAM SKIPS BACK TO THE PUTPOI COMMANDS AND DRAWS ANOTHER LINE. THIS PROGRAM WILL DRAW 11 VERTICAL LINES 500 UNITS HIGH AND 50 UNITS APART. DO YOU SEE WHY 11 LINES ARE DRAWN ? SEE IF YOU CAN DRAW THE PICTURE ON A PIECE OF PAPER AS IF YOU WERE A COMPUTOR AND COULD ONLY FOLLOW THE COMMANDS GIVEN IN THE PROGRAM.

IF YOU UNDERSTAND THE ABOVE PROGRAM AND CAN DRAW THE PICTURE, YOU ARE READY TO TRY EXERCISES 2 AND 3. GO TO THE VT05 KEYBOARD AND TYPE "CALL EX2".

GOOD LUCK !!!

THE COMMANDS AND STATEMENTS PRESENTED IN THIS SECTION WERE:

VARIABLE=EXPRESSION
SKIP EXPRESSION
IF VAR OPR EXP,COMMAND

SECTION 1.5 ..... GRASS STORAGE

BY THIS POINT YOU WILL HAVE COMPLETED THE FIRST THREE EXERCISES. YOU HAVE CREATED PICTURES WITH A POINT BY POINT PLACEMENT METHOD AND BY WRITING PROGRAMS WHICH HAVE PLACED THE POINTS FOR YOU. IN EITHER CASE, THE PICTURES AND THE PROGRAMS YOU HAVE CREATED HAVE SINCE BEEN DESTROYED, AND YOU WILL HAVE TO RECREATE THEM THE NEXT TIME YOU WANT TO USE THEM. IF YOU ARE A NORMAL GRASS USER YOU WILL WANT TO START SAVING THE THINGS YOU HAVE CREATED RATHER THAN RE-CREATING THEM EACH TIME. THE TWO PLACES WHERE YOU MAY NORMALLY STORE PICTURES AND PROGRAMS ARE IN "CORE" AND ON "DISK".

CORE IS THE LOCAL STORAGE IN A COMPUTER. IT IS CALLED CORE BECAUSE OF THE SMALL MAGNETIC CORES (LIKE DONUTS) OF WHICH IT IS MADE. CORE STORAGE IS TEMPORARY STORAGE, AND IS THE PLACE WHERE PROGRAMS AND PICTURES ARE KEPT WHEN THEY ARE BEING CREATED AND USED. WHEN THEY ARE NOT BEING USED, THEY ARE USUALLY STORED ON DISK.

DISK STORAGE, CALLED DISK BECAUSE THAT IS WHAT IT LOOKS LIKE, IS LONG TERM STORAGE. THIS IS WHERE YOU WILL PUT YOUR PICTURES AND PROGRAMS TO KEEP THEM FROM ONE SESSION TO ANOTHER. MANY PICTURES AND PROGRAMS MAY BE STORED ON THE DISK AT THE SAME TIME. THE ONLY RESTRICTION IS THAT NO TWO PICTURES OR PROGRAMS CAN HAVE THE SAME NAME. THIS RULE ALSO APPLIES TO CORE BY THE WAY.

WHEN A PICTURE OR A PROGRAM IS FIRST ENTERED INTO THE COMPUTER IT IS ENTERED INTO CORE STORAGE AND MUST BE COPIED ON TO THE DISK IF YOU WANT IT SAVED. THIS IS DONE WITH THE PUTDSK COMMAND.

SYNTAX: PUTDSK DNAME.EXT

"DNAME" IS THE NAME OF YOUR PICTURE OR PROGRAM (PROGRAMS CAN ALSO BE GIVEN NAMES), AND ".EXT", WHICH STANDS FOR EXTENSION, INDICATES WHAT THE DNAME IS AS FOLLOWS:

.DEC	PICTURE
.MAC	MACRO (PROGRAM)

THERE ARE OTHER EXTENSIONS BUT THESE ARE THE ONLY ONES YOU WILL NEED FOR THE TIME BEING. IF NO EXTENSION IS SUPPLIED, THE COMPUTER WILL TRY TO FIGURE OUT THE PROPER ONE AND WILL SUPPLY IT FOR YOU. FOR EXAMPLE:

PUTDSK PIX.DEC	PICTURE
PUTDSK PROG.MAC	PROGRAM
PUTDSK DRAW	PROGRAM

IF YOU HAVE A PICTURE OR A PROGRAM ON THE DISK AND YOU WANT TO GET IT INTO CORE YOU WILL USE THE GETDSK COMMAND.

SYNTAX: GETDSK UNAME.EXT

"UNAME" AND ".EXT" ARE THE SAME AS IN PUTDSK. HOWEVER IF NO ".EXT" IS SUPPLIED, GRASS WILL ASSUME YOU WANT TO GET A PICTURE, AND IF UNAME IS NOT A PICTURE IT WON'T GET ANYTHING BUT AN ERROR. HERE ARE SOME EXAMPLES:

```
GETDSK PIX.DEC
GETDSK PROG.MAC
GETDSK PIX
```

WHEN PICTURES ARE GETDSK'ED THEY AUTOMATICALLY APPEAR ON THE VECTOR GENERAL SCREEN. WHEN A PROGRAM IS GETDSK'ED IT WILL BE BROUGHT INTO CORE BUT NOTHING ELSE WILL HAPPEN. IT WILL NOT EXECUTE AND IT WILL NOT APPEAR ON THE VT05 VIDEO SCREEN.

PUTDSK'ING AND GETDSK'ING DO NOT CAUSE A PICTURE OR A PROGRAM TO BE REMOVED FROM ONE PART OF THE COMPUTER TO ANOTHER. INSTEAD A COPY IS MADE. THEREFORE ONCE IT IS IN CORE OR ON THE DISK YOUR THING WILL REMAIN UNLESS IT IS ERASED. CORE IS ERASED WHENEVER THE COMPUTER IS TURNED OFF OR RESTARTED AND IS THUS CONSIDERED ONLY TEMPORARY STORAGE.

BECAUSE THE STORAGE SPACE AVAILABLE IN CORE IS MUCH MORE LIMITED THAN ON THE DISK, YOU WILL FIND THAT ON OCCASION, AFTER CREATING SEVERAL PICTURES OR PROGRAMS OR GETTING THEM FROM THE DISK, YOU MAY RUN OUT OF ROOM. IF THIS HAPPENS YOU CAN DELETE SPECIFIC PROGRAMS AND PICTURES WITH THE DELETE COMMAND.

SYNTAX: DELETE ANAME

"ANAME" IS THE NAME OF THE PICTURE OR PROGRAM YOU WISH ERASED. THIS WILL REMOVE ANAME FROM CORE AND YOU MAY THEN USE THE SPACE ANAME OCCUPIED. REMEMBER THOUGH, IF YOU WISH ANAME TO BE SAVED, YOU MUST HAVE A COPY ON THE DISK. IF ANAME CAME FROM THE DISK EVERYTHING IS OK, IF NOT PUTDSK IT BEFORE YOU DELETE IT.

THE DELETE COMMAND CAN ALSO BE USED, IN A SLIGHTLY DIFFERENT VERSION, TO ERASE THINGS FROM THE DISK.

SYNTAX: DELETE/D ANAME.EXT

THE "/D" IS CALLED A SWITCH. WHEN USING THIS VERSION OF THE COMMAND THE EXTENSION (.DEC OR .MAC) MUST BE SUPPLIED.

HERE ARE SOME EXAMPLES OF THE DELETE COMMAND:

```
DELETE PIX
DELETE PROG
DELETE/U PIX.DEC
DELETE/U PROG.MAC
```

REMEMBER, ONCE A PROGRAM OR PICTURE IS ERASED FROM THE DISK AND FROM CORE, IT IS GONE AND WILL HAVE TO BE RETYPED TO BE USED AGAIN.

IF AT ANY TIME YOU ARE NOT SURE WHETHER OR NOT YOUR PICTURE OR PROGRAM IS IN CORE OR ON THE DISK, YOU MAY ASK TO SEE A DIRECTORY OF EITHER. THIS IS DONE BY USING THE DIRCOR OR THE DIRDSK COMMANDS RESPECTIVELY.

```
SYNTAX:     DIRCOR
        OR
SYNTAX:     DIRDSK
```

THESE COMMANDS WILL DISPLAY ON THE VT05 SCREEN A LISTING, BY NAME, OF ALL THE THINGS CURRENTLY STORED IN EACH AREA.

WITH THE COMMANDS PRESENTED IN THIS SECTION AND THE ONES PRESENTED PREVIOUSLY YOU HAVE ALL THE COMMANDS YOU NEED TO CREATE PICTURES POINT BY POINT, AND TO STORE AND RETRIEVE THEM. HOWEVER, IF YOU WISH TO USE PICTURE DRAWING PROGRAMS TO CREATE THE PICTURES FOR YOU, THERE ARE A FEW ADDITIONAL THINGS YOU MUST KNOW ABOUT THE CREATION AND EDITING OF MACROS. THIS INFORMATION IS PRESENTED IN THE FOLLOWING SECTIONS. HOWEVER BEFORE YOU CONTINUE ON IN THE WORKBOOK, I WOULD LIKE YOU TO DO THE NEXT EXERCISE. GO TO THE VT05 KEYBOARD AND TYPE .
CALL EX4.

GOOD LUCK !

THE COMMANDS PRESENTED IN THIS SECTION WERE:

```
PUTDSK DNAME.EXT
GETDSK DNAME.EXT
DELETE ANAME
DELETE/U ANAME.EXT
DIRCOR
DIRDSK
```

SECTION 1.4 ..... MACROS

KINDS OF MACROS BUT IT DOES NOT ALLOW THE SET OF COMMANDS TO BE SAVED AND EXECUTED AGAIN LATER AND BRANCHING MACROS (THOSE WITH CONDITIONAL STATEMENTS OR SKIPS) WILL NOT WORK THIS WAY. THIS BEING THE CASE HOW DOES ONE CREATE AND NAME A MACRO ?

THE PRIMARY METHOD OF CREATING A MACRO IS BY TYPING A NAME, A COLON, AN OPEN BRACKET, A SET OF COMMANDS EACH FOLLOWED BY A CARRIAGE RETURN, AND A CLOSE BRACKET FOLLOWED BY A CARRIAGE RETURN

```
* MNAME:<COMMAND
+ COMMAND
+ COMMAND
    ETC.
+ COMMAND>
*
```

THE STARS "*" AND THE PLUS SIGNS "+" ARE NOT TYPED IN BY YOU BUT ARE DISPLAYED BY GRASS AND INDICATE THE LEVEL OF GRASS OPERATION. IN STAR LEVEL GRASS WILL ACCEPT COMMANDS FOR IMMEDIATE EXECUTION. IN PLUS LEVEL GRASS WILL WAIT FOR ADDITIONAL INFORMATION (IN THIS CASE THE REST OF YOUR COMMANDS) BEFORE EXECUTING ANY COMMANDS.

MNAME IS THE NAME OF YOUR MACRO (6 LETTERS OR LESS), COMMAND IS ANY VALID GRASS COMMAND STATEMENT. FOLLOWING EACH CARRIAGE RETURN, GRASS WILL PRINT A PLUS SIGN UNTIL IT SEES THE CLOSE BRACKET, AT WHICH TIME IT WILL RESPOND WITH A "*", WHICH SHOWS GRASS IS AGAIN READY TO ACCEPT COMMANDS FOR EXECUTION.

YOUR MACRO IS NOW STORED IN CORE UNDER THE NAME YOU HAVE GIVEN IT, AND CAN BE EXECUTED WITH THE DO COMMAND.

SYNTAX: DO MNAME

WHERE "MNAME" IS THE NAME OF THE PROGRAM YOU WISH TO EXECUTE. THE DO COMMAND WILL CAUSE A PROGRAM WHICH IS ALREADY IN CORE TO EXECUTE, OR IF THE PROGRAM IS NOT IN CORE, BUT ON THE DISK, DO WILL GET THE PROGRAM INTO CORE AND THEN EXECUTE IT.

FOR EXAMPLE YOU MIGHT TYPE IN THE FOLLOWING SEQUENCE:

```
* DRAW:<OPEN LINE
+ PUTPOI 0,0,0,1
+ PUTPOI 1000,0,0,0
+ CLOSE>
* DO DRAW
```

WHEN GRASS SEES THE CARRIAGE RETURN AFTER THE "DO DRAW" COMMAND STATEMENT, THE MACRO WILL EXECUTE AND DRAW A LINE CALLED "LINE".

DON'T CONFUSE THE CLOSE COMMAND WITH THE ENDING OF THE MACRO. THIS COMMAND IS NECESSARY TO END THE PICTURE, NOT THE MACRO. A MACRO HAS NO EXPLICIT END STATEMENT. GRASS WILL EXECUTE THE COMMANDS IN A MACRO UNTIL THEY RUN OUT AND THEN IT WILL RETURN TO STAR LEVEL AND AWAIT FURTHER COMMANDS.

ONE WOULD PROBABLY NOT WRITE A MACRO TO DRAW A SIMPLE ONE LINE PICTURE. INSTEAD, THE PICTURE WOULD BE OPENED, THE POINTS WOULD BE PLACED, THE PICTURE WOULD BE CLOSED AND THEN STORED ON DISK UNDER THE PICTURE NAME. HOWEVER, IT IS NOT DIFFICULT TO THINK OF A MACRO WHICH WOULD CREATE A PICTURE THAT WOULD REQUIRE MORE STORAGE SPACE THAN THE MACRO (SOME OF THE MACROS IN EXERCISE 3 WERE LIKE THIS). IN THIS CASE IT MIGHT BE WISE TO STORE THE MACRO AND NOT THE PICTURE.

YOU MAY FIND THAT WHEN YOU FIRST EXECUTE A MACRO, GRASS RESPONDS WITH AN ERROR. THE ERROR INDICATION SHOULD LOOK AS FOLLOWS:

```
??REPEAT OF LINE IN ERROR
??          ^
??ERROR #1
FIX ERROR & TYPE 'RESUME' OR CONTROL-C
#
```

THE FIRST LINE OF THE ERROR INDICATION WILL REPEAT THE LINE IN YOUR MACRO IN WHICH THE ERROR OCCURRED. THE SECOND LINE WILL CONTAIN A POINTER WHICH WILL INDICATE, IF POSSIBLE, WHERE THE ERROR OCCURRED IN THE LINE. THE THIRD LINE WILL TELL YOU THE ERROR NUMBER AND THE FOURTH LINE WILL TELL YOU WHAT YOU CAN DO ABOUT IT. THE FIFTH LINE WILL JUST BE A NUMBER SIGN "#".

THERE ARE ABOUT 200 POSSIBLE ERROR NUMBERS THAT CAN BE PRINTED BY GRASS. THERE IS A LISTING POSTED ON THE WALL, BUT IF YOU WISH YOU CAN TYPE A QUESTION MARK "?" FOLLOWED BY A CARRIAGE RETURN AND GRASS WILL PRINT OUT AN ERROR MESSAGE. IF YOU CAN FIX THE ERROR DO SO THEN HIT THE CARRIAGE RETURN AND TYPE RESUME. IF THE ERROR IS FIXED YOUR MACRO WILL CONTINUE TO RUN. IF YOU CANNOT FIX THE ERROR, HOLD DOWN THE "CTRL" KEY AND TYPE "C", AND GRASS WILL RETURN TO THE STAR LEVEL.

WHEN YOU CREATE MACROS, WHETHER THEY BE TO DRAW PICTURES OR DO OTHER THINGS, AND ERRORS OCCUR, YOU MAY HAVE TO CHANGE SOME OF THE COMMAND STATEMENTS. TO PREVENT YOU FROM HAVING TO RETYPE THE ENTIRE MACRO, SOMETHING CALLED AN EDITOR HAS BEEN PROVIDED. THIS ALLOWS YOU TO EDIT YOUR MACRO, WHICH IS PROBABLY WHAT YOU THOUGHT IT WOULD DO. TO LEARN ABOUT THE GRASS EDITOR PROCEED TO THE NEXT SECTION.

SECTION 1.5 ..... EDITING

AN EDITOR IS SOMETHING WHICH ALLOWS COMPUTER PROGRAMS, STORED IN CORE OR ON DISK, TO BE CHANGED. SINCE YOU CAN NOT PHYSICALLY CHANGE A PROGRAM ONCE IT IS IN THE COMPUTER, YOU MUST DO IT ELECTRONICALLY.

TO ENTER THE EDITING MODE IN GRASS, YOU MUST FIRST HAVE A MACRO TO EDIT. IT NEED NOT, HOWEVER, BE COMPLETE. THUS, IF YOU DISCOVER A MISTAKE IN A MACRO WHILE YOU ARE FIRST TYPING IT IN, YOU MAY USE THE CLOSE BRAKET TO GET BACK TO STAR LEVEL, ENTER THE EDITING MODE, CORRECT THE MISTAKE,AND THEN FINISH TYPING IN THE MACRO. THE EDITING MODE IS ENTERED BY TYPING THE EDIT COMMAND:

SYNTAX: EDIT MNAME

"MNAME" IS THE NAME OF THE MACRO YOU WISH TO EDIT. IT MAY EITHER BE IN CORE OR ON THE DISK. WHEN THIS COMMAND IS TYPED, GRASS WILL RESPOND WITH A QUESTION MARK "?" AND WAIT FOR YOUR REPLY. YOUR REPLY, OF COURSE, WILL DEPEND ON WHAT YOU WANT TO DO TO YOUR MACRO.

THERE ARE FOUR THINGS WHICH THE EDITOR WILL ALLOW YOU TO DO TO A MACRO:

1. DISPLAY IT, SO THAT YOU CAN SEE WHAT YOU HAVE AND WHAT NEEDS TO BE CHANGED.
2. INSERT OR ADD A LINE TO THE MACRO.
3. DELETE OR REMOVE A LINE FROM THE MACRO.
4. CHANGE PART OF A LINE THAT IS IN THE MACRO.

IF YOU RESPOND TO THE QUESTION MARK BY TYPING A CARRAGE RETURN (FROM NOW ON <CR> WILL STAND FOR CARRAGE RETURN), THE EDITOR WILL DISPLAY THE MACRO, LINE BY LINE, UNTIL IT GETS TO THE END. EACH LINE WILL HAVE A NUMBER; 10 FOR THE FIRST LINE, 20 FOR THE SECOND, 30 FOR THE THIRD, AND SO ON. NOTE THAT THE FIRST LINE WILL NOT DISPLAY THE NAME OF THE MACRO, THE COLON, OR THE OPEN BRACKET, AND THE LAST LINE WILL NOT DISPLAY THE CLOSE BRACKET. THESE ARE ONLY USED WHEN THE MACRO IS BEING NAMED.

IF YOU DO NOT WISH THE ENTIRE MACRO TO BE DISPLAYED "100<CR>" WILL DISPLAY ONLY LINE 100, AND "200,300<CR>" WILL DISPLAY LINES 200 THROUGH 300. THE FOLLOWING WILL INSERT, DELETE, OR CHANGE THE MACRO AS DESCRIBED:

"51 OPEN PIX<CR>" WILL INSERT THE COMMAND "OPEN PIX" BETWEEN LINES 50 AND 60.
"100 DO DRAW<CR>" WILL REPLACE LINE 100 WITH THE COMMAND "DO DRAW" OR WILL ADD LINE 100 TO A MACRO IN WHICH LINE 90 WAS THE LAST LINE.
"230=<CR>" WILL DELETE LINE 230 FROM A MACRO.
"60/PUTPIO/PUTPOI/<CR>" WILL CHANGE "PUTPIO" TO "PUTPOI" IN LINE 60.

NOTE THAT WHENEVER A LINE IS INSERTED OR DELETED ALL THE FOLLOWING LINE NUMBERS ARE IMMEDIATELY CHANGED. IT IS A GOOD IDEA, THEREFORE, TO DISPLAY THE LINES AFTER EACH INSERT OR DELETE.

ONCE YOU HAVE FINISHED EDITING YOUR MACRO YOU CAN GET GRASS BACK TO STAR LEVEL BY TYPING A CONTROL L (HOLD DOWN THE CTRL KEY AND TYPE L). GRASS WILL RESPOND WITH A "*" AND IS NOW READY TO ACCEPT COMMANDS.

IF YOUR MACRO WAS PREVIOUSLY STORED ON DISK, THE DISK COPY WILL NOT BE THE SAME AS THE NEW EDITED VERSION IN CORE. TO PUT THE NEW VERSION ON DISK USE THE PUTDSK COMMAND WITH THE "/O" SWITCH:

SYNTAX: PUTDSK/O MNAME

THIS WILL DO TWO THINGS. FIRST IT WILL TAKE THE VERSION OF THE MACRO ALREADY ON DISK AND GIVE IT A NEW EXTENSION ".BAK" (WHICH STANDS FOR BACKUP). SECOND IT WILL STORE THE NEW VERSION OF THE MACRO ON THE DISK WITH THE SAME NAME BUT WITH THE ".MAC" EXTENSION. THIS PROTECTS THE ORIGIONAL VERSION OF THE MACRO IN CASE YOU NEED IT SOMETIME.

YOU MAY ALSO SAVE THE ORIGIONAL VERSION OF YOUR MACRO BY GIVING THE EDITED VERSION A NEW NAME. THIS IS DONE WITH THE RENAME COMMAND:

SYNTAX: RENAME ANAME1,ANAME2

THIS COMMAND WILL CAUSE A PICTURE OR A MACRO IN CORE WITH THE NAME "ANAME1" TO HAVE A NEW LABLE, "ANAME2". YOU CAN THEN PUT ANAME2 ON THE DISK AND THUS HAVE BOTH VERSIONS.

IN ADDITION TO CREATING PICTURES POINT BY POINT, YOU SHOULD NOW BE ABLE TO CREATE PICTURE DRAWING MACROS TO DO SOME OF THE WORK FOR YOU. YOU HAVE EVERYTHING YOU NEED, BY WAY OF GRASS COMMANDS, TO DO THIS ON YOUR OWN. SO GO AHEAD AND BEGIN CREATING.

TO LEARN MORE ABOUT WHAT YOU CAN DO WITH THE PICTURES YOU CREATE, AND LEARN HOW TO DO IT, GO ON TO PART 2.

THE COMMANDS PRESENTED IN THIS SECTION WERE:

EDIT MNAME
PUTDSK/O MNAME
RENAME ANAME1,ANAME2