Navigating INFO

presented by:
Tim Haithcoat
University of Missouri Columbia

with materials from:
Environmental Systems Research Institute

Section Five

Programming

Odd & Even Program Sections

ODD SECTION

CALC MAT_COST = MILES * $NUM1

CALC LBR_COST = MILES * $NUM2

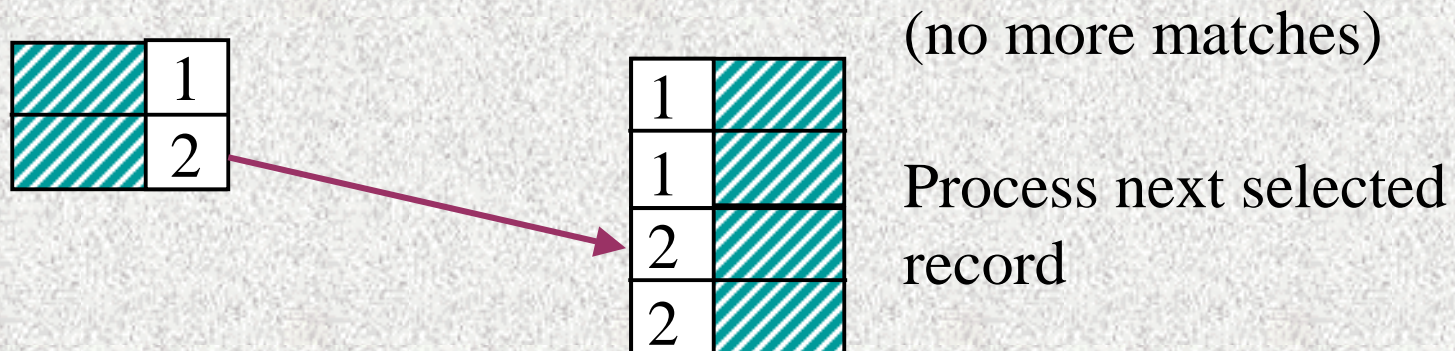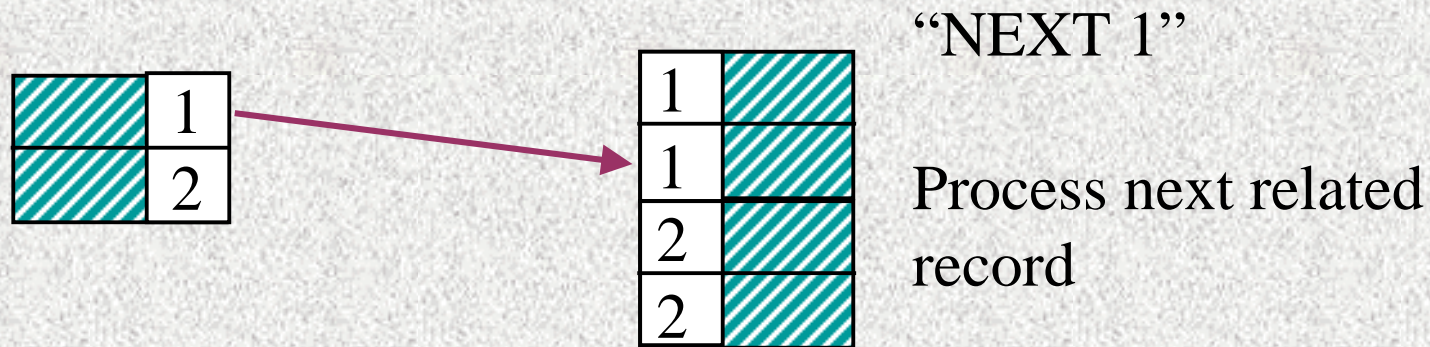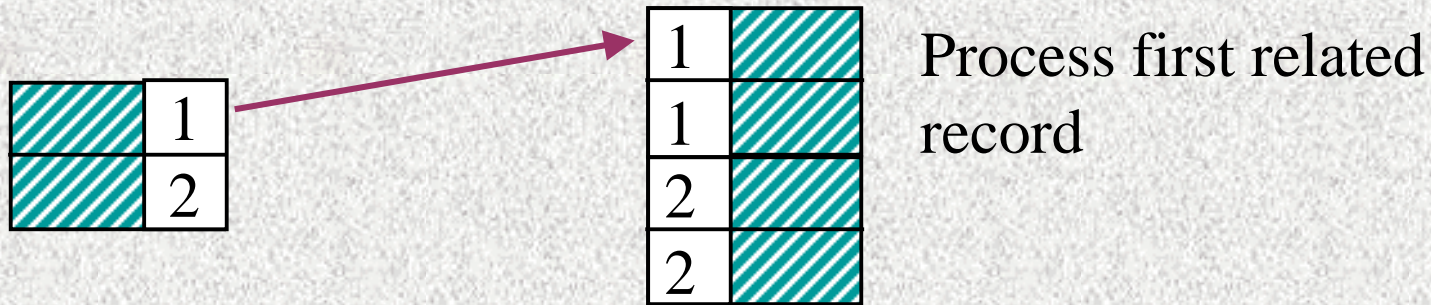CALC TOTAL_COST = MAT_COST + LBR_COST

Total bytes read =

  3* $NOSEL * Width

PROGRAM SECTION EVEN

CALC MAT_COST = MILES * $NUM1

CALC LBR_COST = MILES * $NUM2

CALC TOTAL_COST = MAT_COST + LBR_COST

PROGRAM SECTION ODD

EVEN SECTION

Total bytes read =

  $NOSEL * Width

NEXT PROCESSING

Process first related record

"NEXT 1"

Process next related record

(no more matches)

Process next selected record

NEXT PROCESSING

Process last related record

(no more matches)

Process next selected record

"First 1"

Process first related record
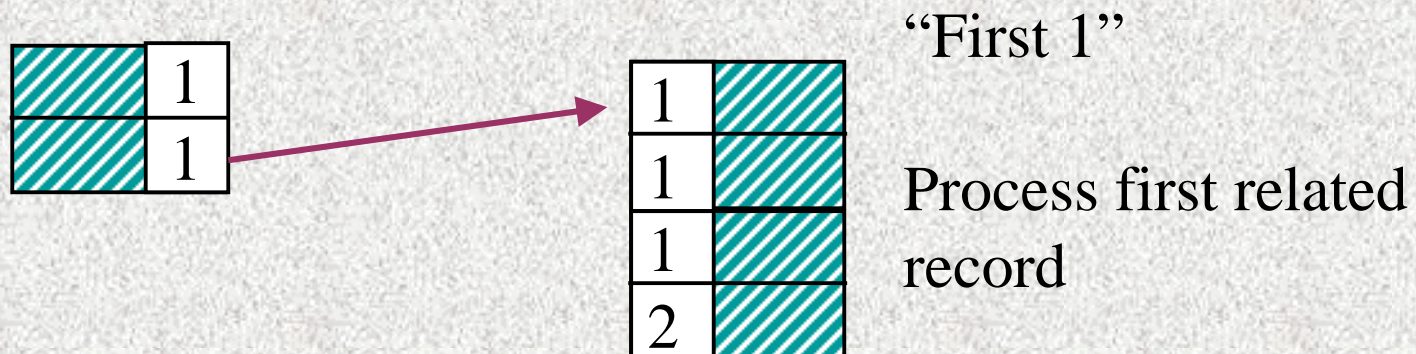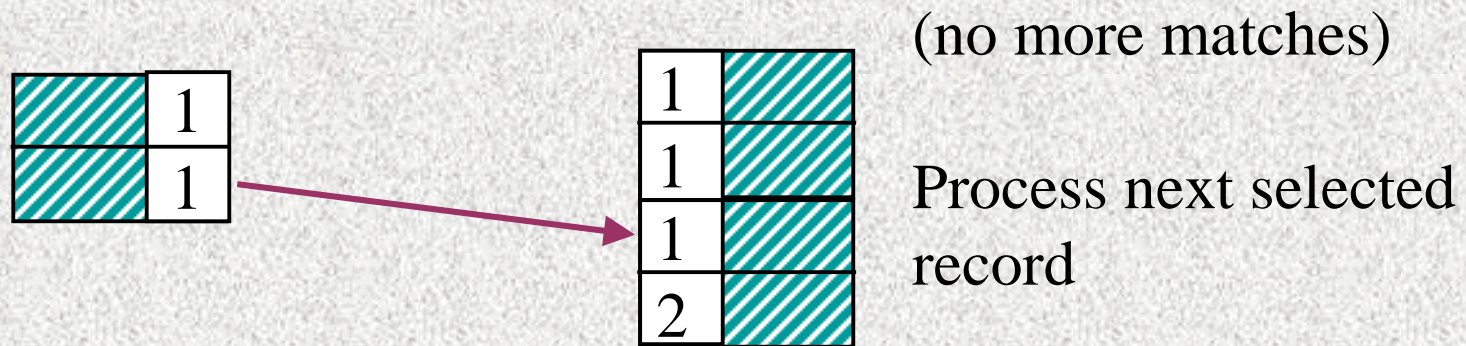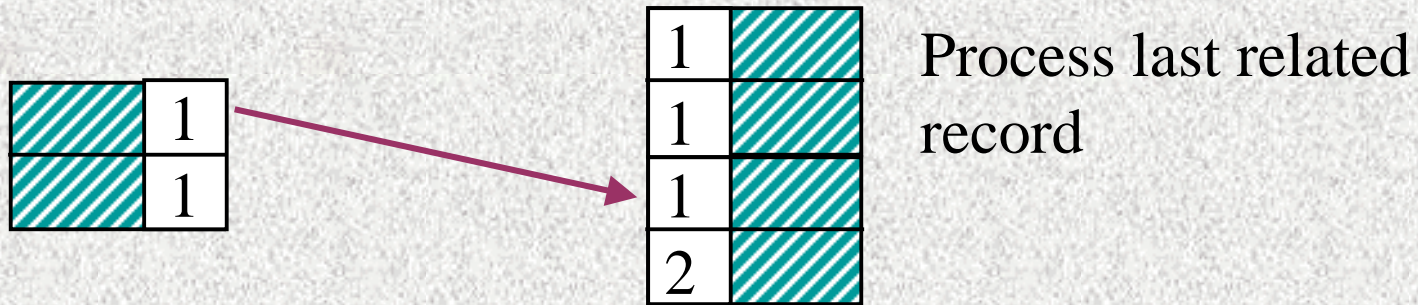
Odd & Even Program Sections

- ## GOTO…LABEL

  - Cannot branch to a label that is contained in another program section

- ## DO…DOEND

  - A DO loop may not cross program sections

- ## IF…ENDIF

  - An IF block cannot cross program sections

Odd & Even Program Sections
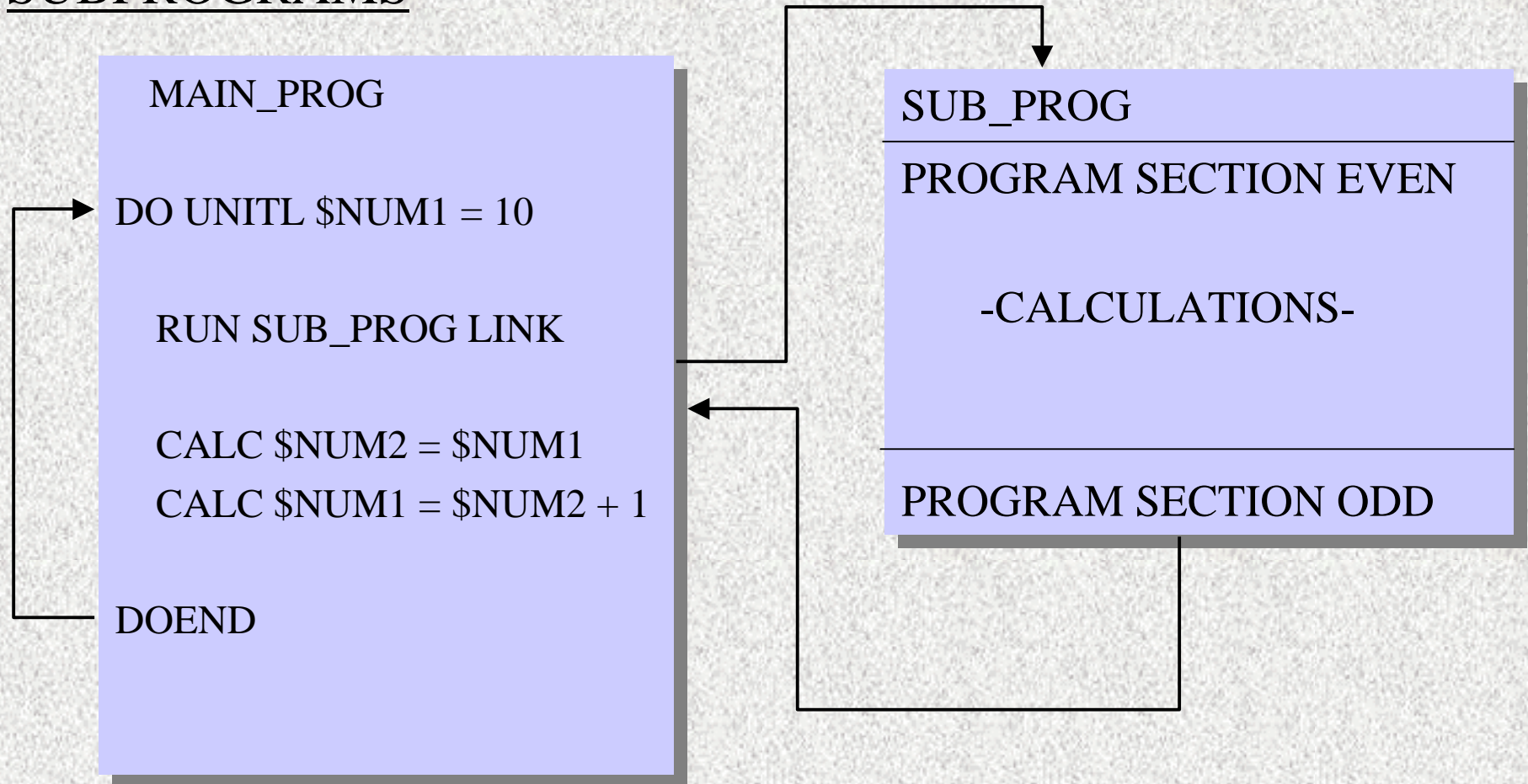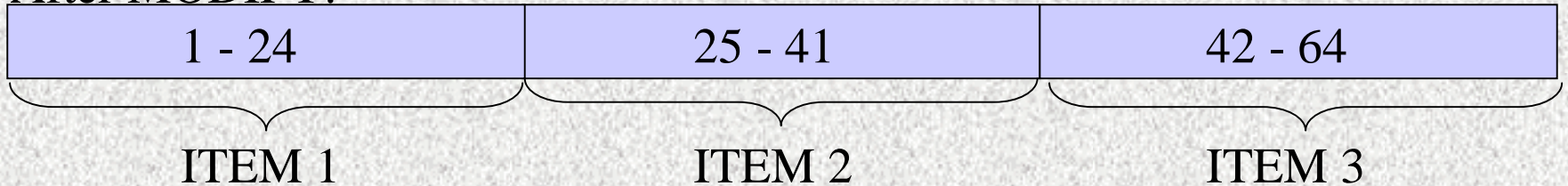
SUBPROGRAMS

MAIN_PROG

DO UNITL $NUM1 = 10

   RUN SUB_PROG LINK

   CALC $NUM2 = $NUM1
   CALC $NUM1 = $NUM2 + 1

DOEND

SUB_PROG

PROGRAM SECTION EVEN

   -CALCULATIONS-

PROGRAM SECTION ODD

After COMPILE:

1 - 20	21 - 27	38 - 60
ITEM 1	ITEM 2	ITEM 3
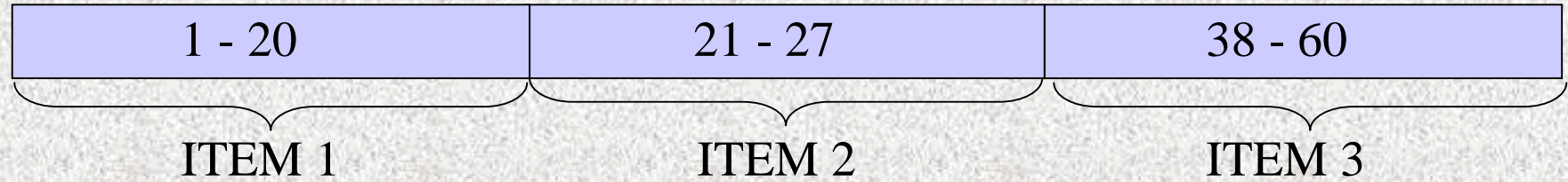
COMPILE records item columns in program OBJ file.

After MODIFY:

1 - 24	25 - 41	42 - 64
ITEM 1	ITEM 2	ITEM 3

Item positions recorded by COMPILE no longer match item definitions.

After COMPILE:

1 - 24	25 - 41	42 - 64
ITEM 1	ITEM 2	ITEM 3
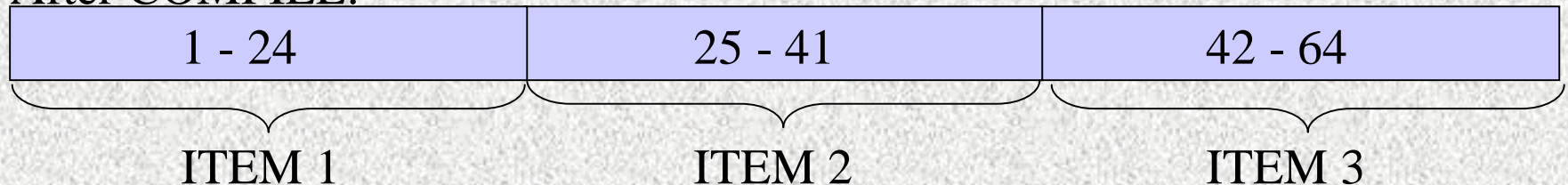
COMPILE updates item positions to match new file definition.

Compiling and Decoding

- Tells INFO to compile code again

- Performs no immediate action

- Info editor automatically sets decode flag

- Always use decode when debugging

Debugging Techniques

- Test Files

- USE $NM

- Run with LIST

- Displaying Checkpoints

- Breaking from a Program (Prime Users OnlY)

Editing INFO Programs

- INFO CHANGE COMMAND

- SRC FILES

- COMI FILES

- AML PROGRAMS

Editing INFO Programs

The following AML is an example:

```
/* MK_CALC-IDS.AML
/*
/* Recalculate user Ids for a specified
/* polygon coverage
/*
&Args Cov
&If Not [Exists %Cov% - poly) &Then ~
&Return Coverage %Cov% not found or has no topology.

&SET Have_prog := {Exists CALC-IDS - INFO]

&Data Arc Info
&If %Have_prog% &Then &Do
Erase CALC-IDS
Y
&End
PROGRAM CALC-IDS
SELECT %Cov%.PAT
CALC %COV%-ID = %COV%# - 1
END
    /* This line blank to end program.
RUN CALC-IDS
ERASE CALC-IDS
Y
QUIT,STOP
&End
&Return
```
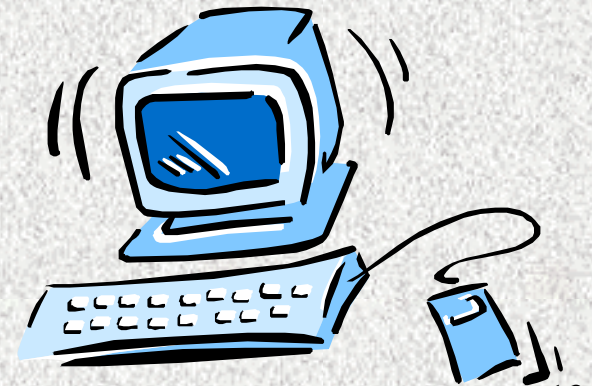
Of course, we didn't really need a program to do this calculation, but it illustrates the point.

Running Programs with LINK

- Returns to calling program

- Do not use END

- Automatically compiled with calling program

- Use as subroutines

- Use link to run menu options

Menu Files and Programs

AN EXAMPLE:

```
PROGRAM MENU.PG
FORMAT $NUM1,22,I                                  Accepts option number from user.
CALC $NUM1 = -1
SELECT SCHED RO
DO UNTIL $NUM1 GE O and $NUM1 LE $NOREC
    DIS =
    DIS '                                          Option   Action'
    DIS ' '
    DIS '                                          ',$RECNO,' -',TEXT
    DIS ' '
    DIS '    Enter the number of your choice, or <CR> to quit: ',=
    ACCEPT $NUM1
DOEND
IF $NUM1 = 0
    QUIT/STOP
ELSE
    RESELECT BY $RECNO = $NUM1
    EXECUTE RUN_COM
    RUN MENU.PG
ENDIF
```

Menu Files and Programs

- Options kept in an INFO file

- Menu program uses EXECUTE

- Compact, portable and modifiable

- Variations

  - HELP program

  - Control of access