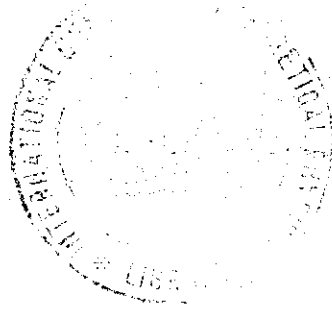




INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION



INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

34100 TRIESTE (ITALY) - P.O. B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 224221/2/3/4/5/6
CABLE: CENTRATOM - TELEX 480392-1

SMR/90 - 11

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

7 September - 2 October 1981

PHYSICIST FORTRAN

A. van DAM

Computer Science Department
Brown University
Providence, R.I. 02912
USA

These are preliminary lecture notes, intended only for distribution to participants. Missing or extra copies are available from Room 230.

PHYSICIST FORTRAN

-1-

DIMENSION C...

I = 1

IF ALPHA GT 0 GOTO 430 (where is that?)

Z1Q = ---

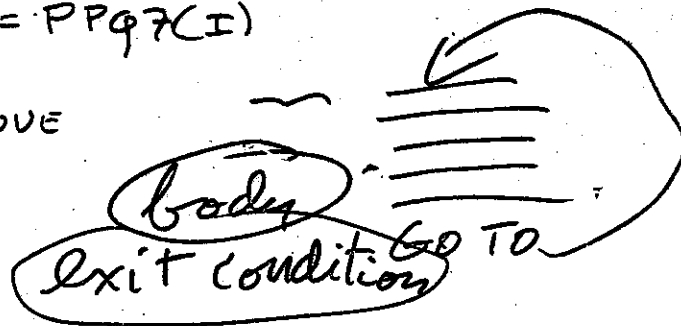
280 ← how can I get here?

DO 9910 I = 1 TO N

ALPHA = PPQ7(I)

9990 CONTINUE

hand simulate



UNREADABLE BY MAN OR BEAST

1. NO COMMENTS

2. NO INDENTATION

3. NO REASONABLE VARIABLE NAMES

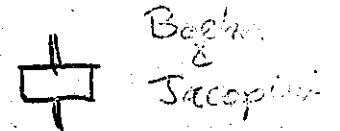
4. TOO LITTLE MODULARIZATION

* 5. CAN'T FOLLOW/PREDICT FLOW OF CONTROL
LABELS, GOTO'S, NO STRUCTURED CONSTRUCTS

STRUCTURED FLOW-OF-CONTROL

-2-

• 1-in, 1-out BB

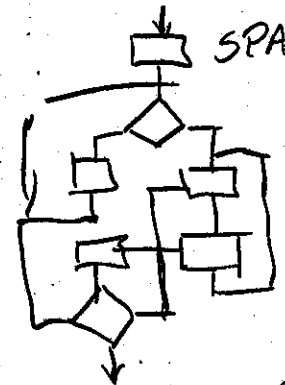


• predictable



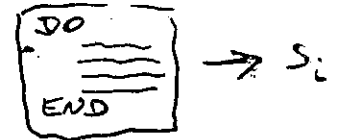
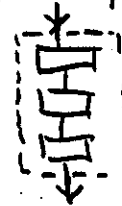
that's not

SPAGHETTI



• functionally complete set: brackets

sequence



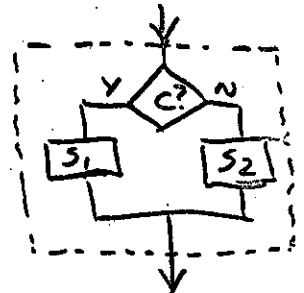
test

2) IF C

THEN S1

[ELSE S2]

ENDIF

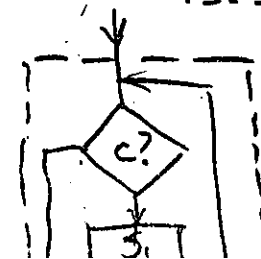


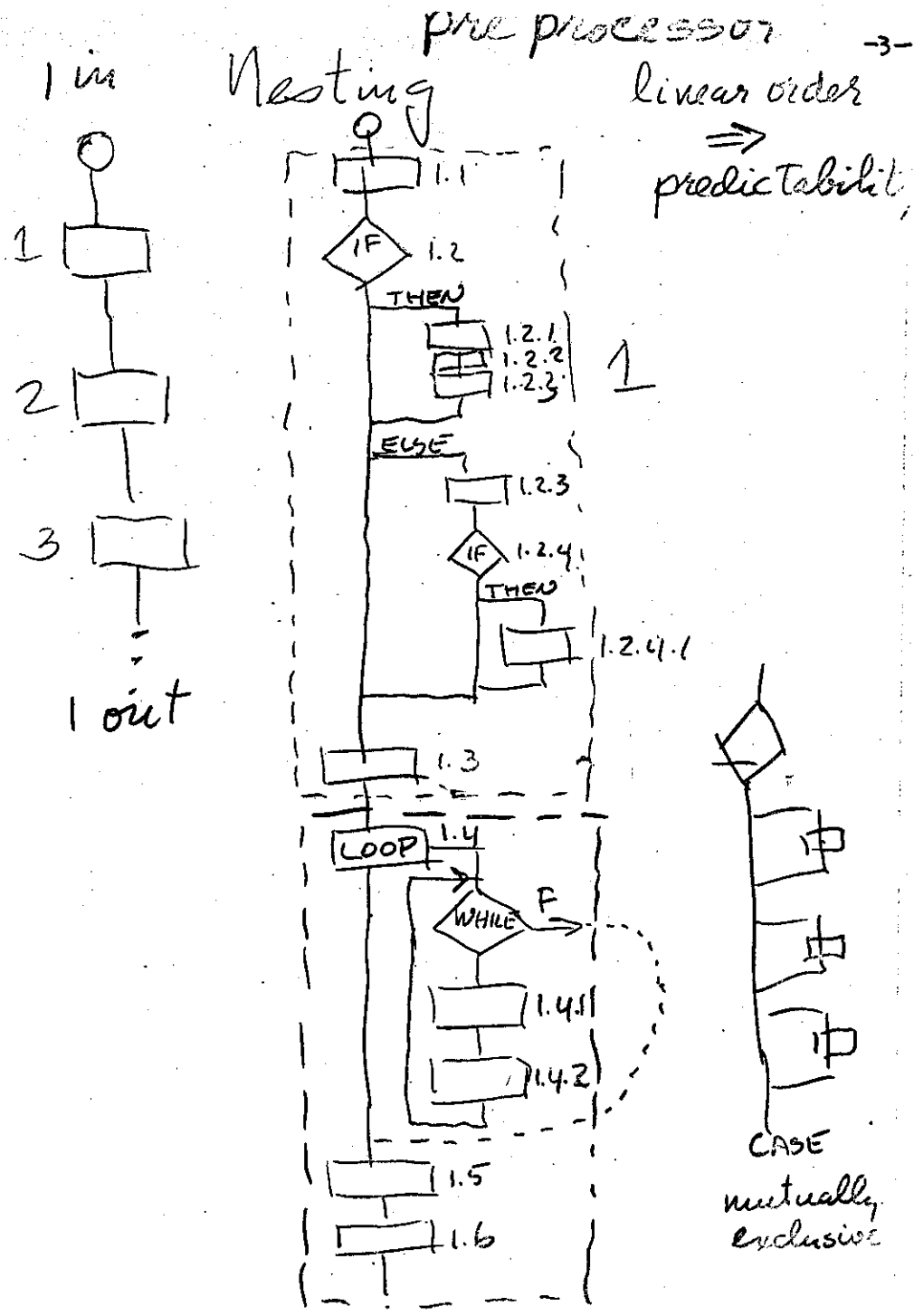
loop

3) DO WHILE C

S

ENDWHILE





Top Down Stepwise Refinement -4-

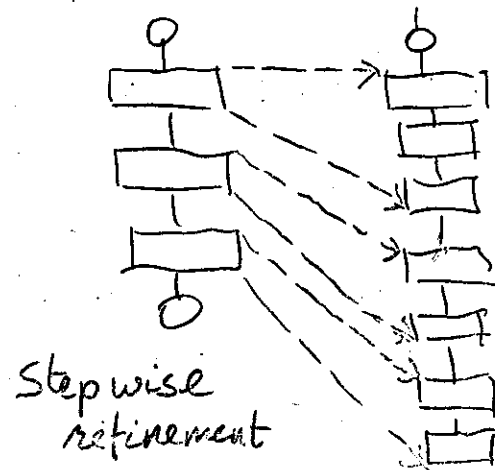
LO: Input a list of non-negative integers: a) call a processing routine for positive integers, an error routine for zero values b) output a count of positive values

Q1: How is list terminated?

A1: First negative number

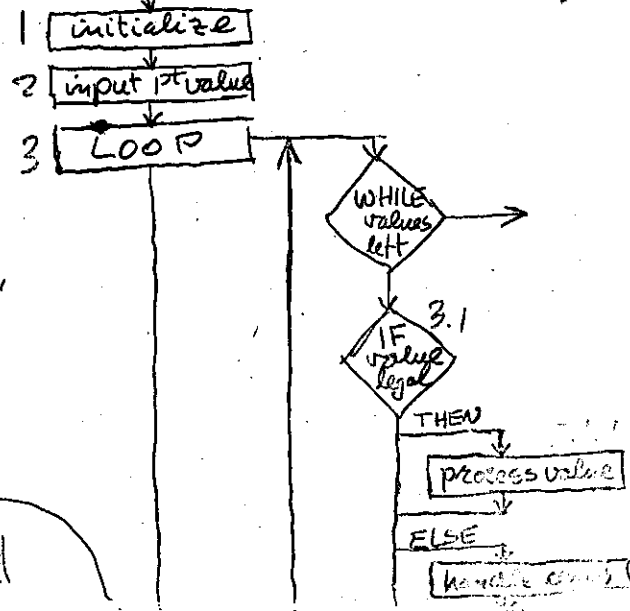
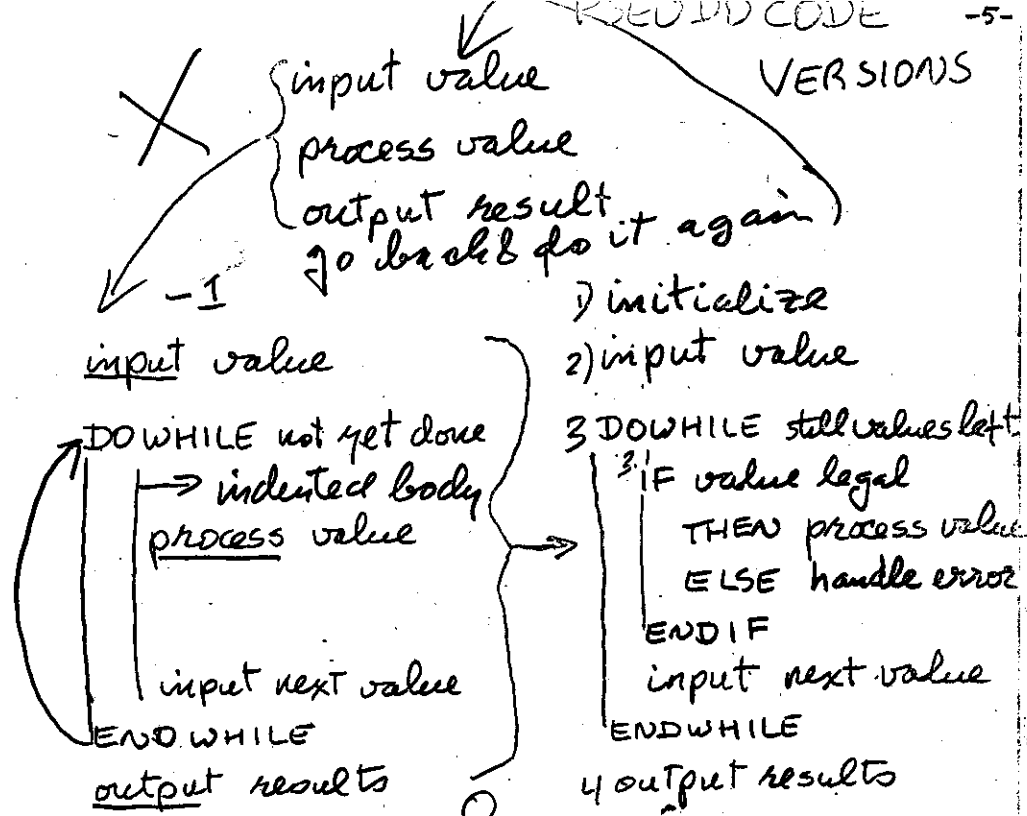
Q2: Process "on-the-fly" or store(array)?

A2: on-the-fly. (Histogram ⇒ array)
∴ no data structure, just a count



pseudo code $\triangleq$

English, maths
embedded in
STRUCTURED
FLOW OF CODE



Q/D
RULE

"WHAT IF...?"
POSSIBLE!!!

MPL implementation
explanatory header/block comment
count = 0
CALL read (value)
1 step/thing
at a time

DOWHILE value GE 0
IF value GT 0
THEN DO
count = count + 1
CALL process (value)
END
ELSE
CALL error routine
ENDIF
CALL read (value)
END! WHILE
CALL display (count)

For assembly language coding,
take each statement in turn, use
programming reference manual

BE PARANOID

MPL/FORTRAN $\rightarrow$ assembly language

ALPHA = BETA + RATE

MPL/FORTRAN COMPILER

LDA A BETA
ADD A RATE
STA A ALPHA

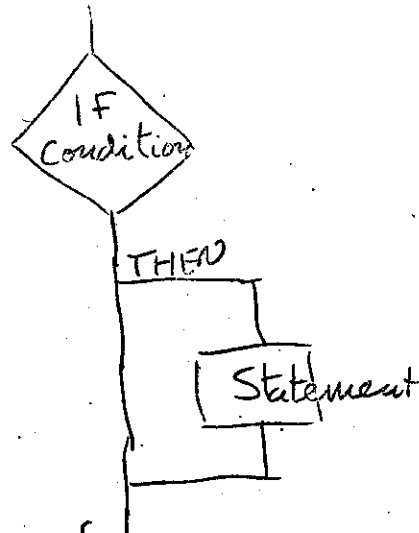
$A \leftarrow (BETA)$
 $A \leftarrow (A) + (RATE)$
 $ALPHA \leftarrow (A)$

ASSEMBLER

B6
12
F4
BB
03
E2
B7
04
24

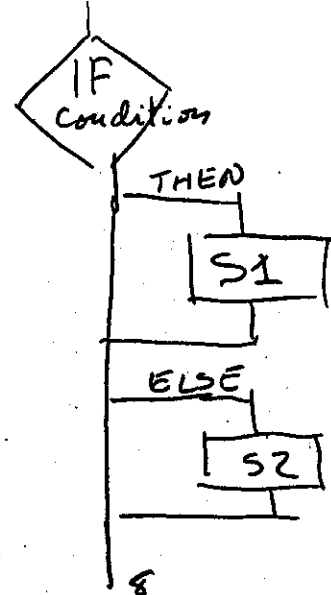
Symbol table

12F4 BETA



pseudocode

label	OP code	operand	Comments
1	8	16	32
	LDA	ALPHA	IF alpha LE 10
	CMP	#10	
	BHI	ENDIF1	.. THEN
	LD		<then code>
ENDIF1	EQU	*	END IF
9950	CONTINUE		



Mutually exclusive

