Advanced Techniques / COBOL

All rights reserved to D.Maymone

תוכן עניינים

- מבוא
- Structured/Functional IF
- Tree Charts
- מערכים
- אופציות מיוחדות בשפת COBOL
- מודלים לתכנות
- טבלאות החלטה
- State Diagrams
- טבלאות מצבים
- שיטות גישה מתקדמות במערכים
- תרגילים

All rights reserved to D.Maymone

מבוא

- תכונות של "תוכניתן טוב"
- מדדי איכות תוכנה
- "משבר התוכנה"
- היסטוריה

All rights reserved to D.Maymone

תכונות של תוכניתן "טוב"

- כותב תוכניות "טובות"
- הספק גבוה
- מוצא מהר באגים
- מוצא פתרונות פשוטים לבעיות מורכבות

All rights reserved to D.Maymone

Measurements of Quality Programming

- Integrity
- Efficiency
- Flexibility
- General
- Maintenance and Documentation
- Simplicity
- Modularity
- Standards

All rights reserved to D.Maymone

משבר התוכנה

- הפער בין החומרה והתוכנה
- עלות התוכנה
- מורכבות
- עלות התחזוקה
- דרישות לשינויים
- החלפת/ניוד עובדים
- תיעוד
- Tests

All rights reserved to D.Maymone

היסטוריה

- 59 – COBOL
- 65 – Dijkstra: the software quality is inversely proportional to the number of '*GO Tos*'
- 69 – Dr. Mills – Super programmers ==> 1:30
- 72 – The investment in testing grows exponentially with the size of programs
- 72 – Bohem-Jacopini: every program should be coded by 3 basic structures only
- 86 – Object Oriented Programming

All rights reserved to D.Maymone

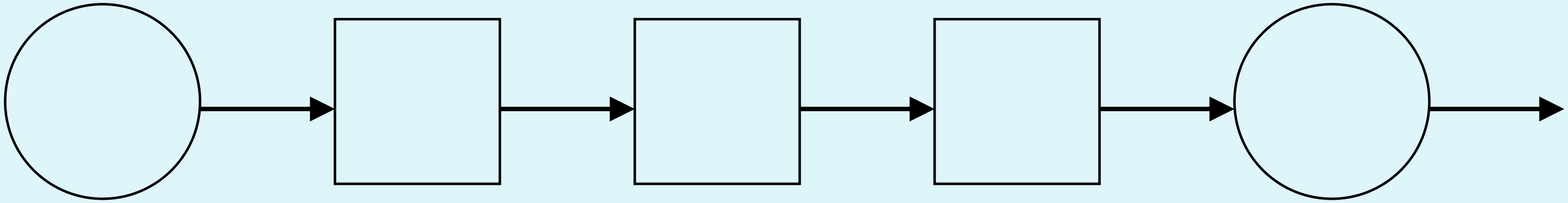
היסטוריה (המשך)

1970	Structured Programming
1975	Structured Design
1978-1980	Structured Analysis
1978-1980	Data Modeling
1980	4GL
1986	OOP
1986	CASE
1990	OOAD
1997	UML

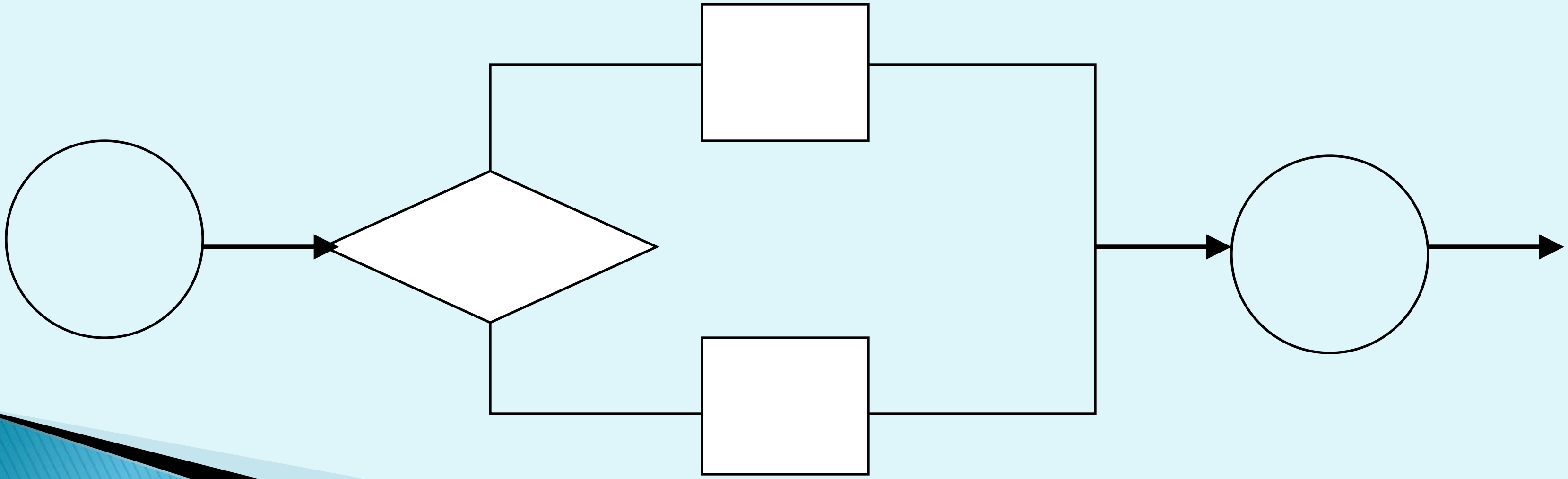
All rights reserved to D.Maymone

3 מבנים בסיסיים

- Sequence

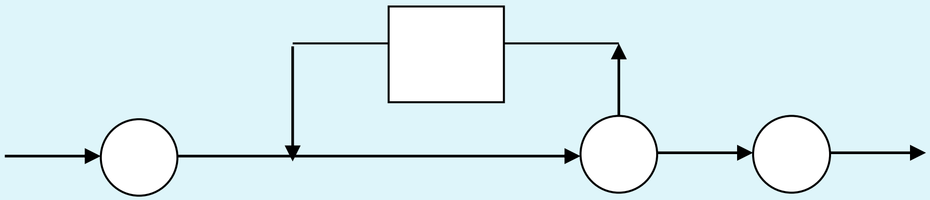


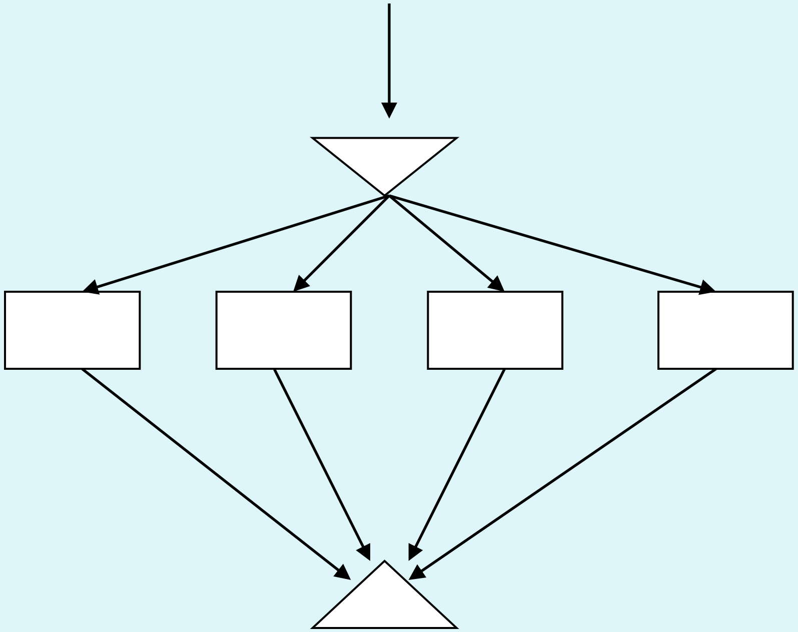
- IF-THEN-ELSE



All rights reserved to D.Maymone

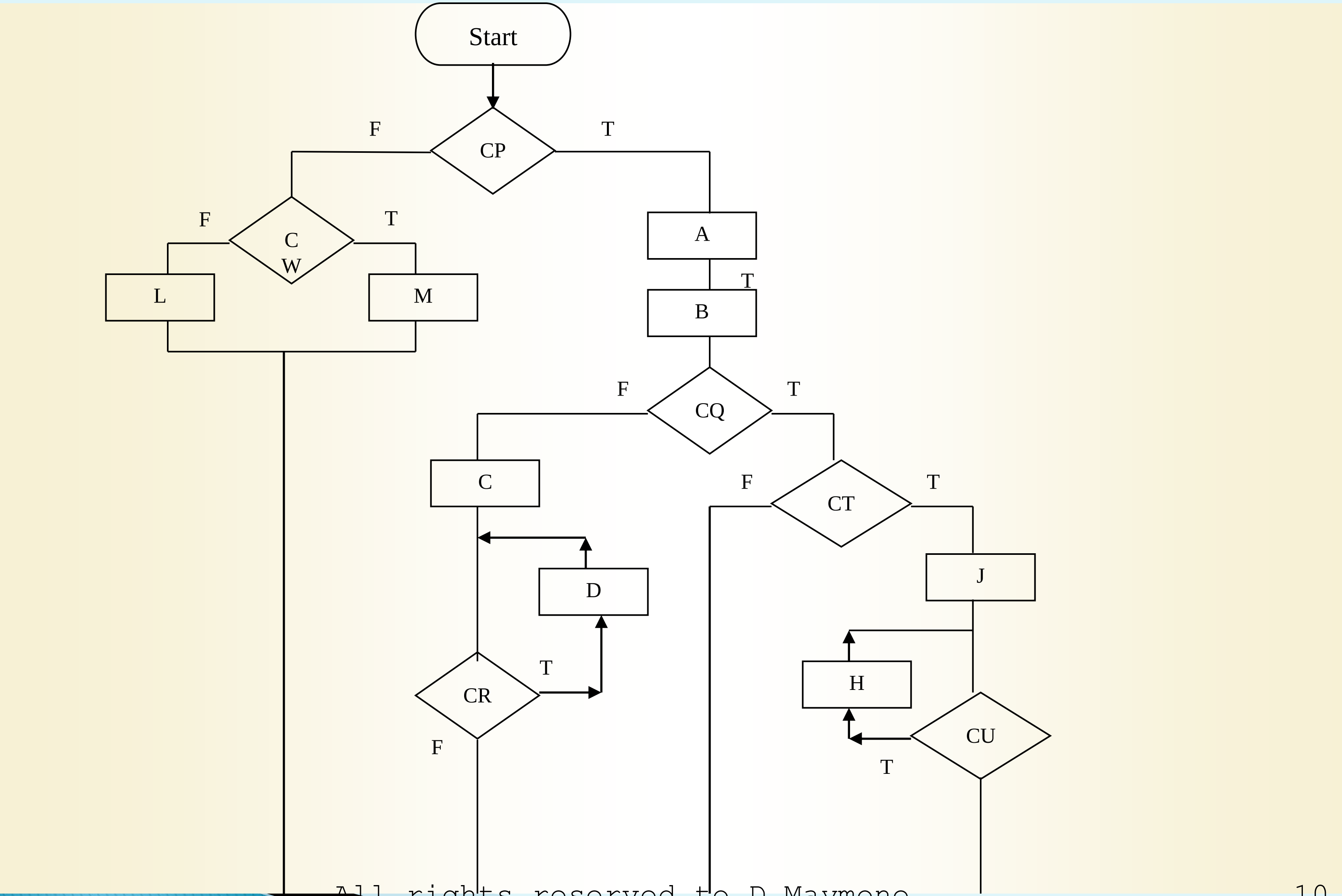
3 מבנים בסיסיים (המשך)





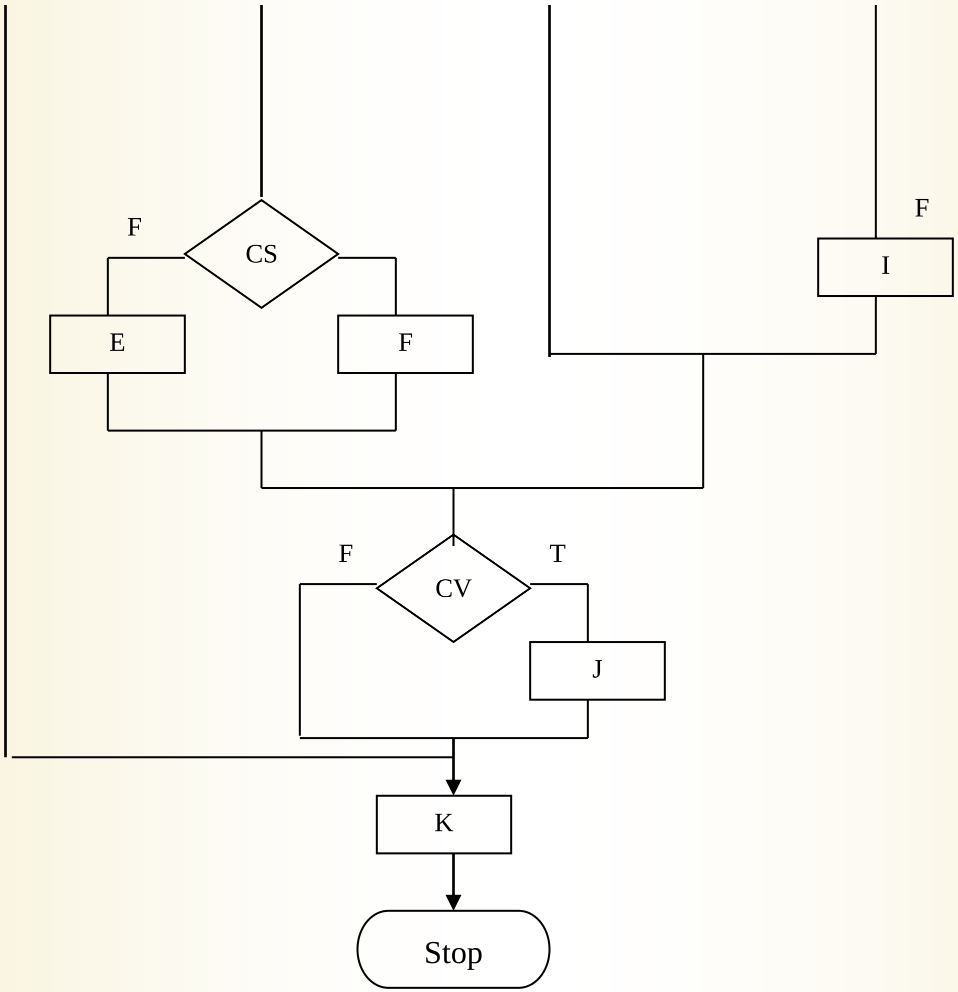
All rights reserved to D.Maymone

Structured IF



All rights reserved to D.Maymone

Structured IF (cont'd)



All rights reserved to D.Maymone

Structured IF – Pseudocode

```
If  CP
      Call  Proc-A
      Call  Proc-B
      If  CQ
            If  CR
                  Call  Proc-C
                  Call  Proc-D
            End-If
      Else
            Call  Proc-E
            If CS
                  Call  Proc-F
            Else
                  Call  Proc-G
            End-If
      End-If
      If CT
            Call Proc-H
      End-If
Else
      If CU
            Call Proc-I
      Else
            Call Proc-J
End-If
Call Proc-K
```

All rights reserved to D.Maymone

Functional IF

Condition / Function	P	Q	R	S	T	U
A	T					
B	T					
C	T	T	T			
D	T	T	T			
E	T	F				
F	T	F		T		
G	T	F		F		
H	T				T	
I	F					T
J	F					T

All rights reserved to D.Maymone

משפט ה- Go To

- Do not use goto
- “The GOTO must GO”
- “The ELSE must GO TOO”

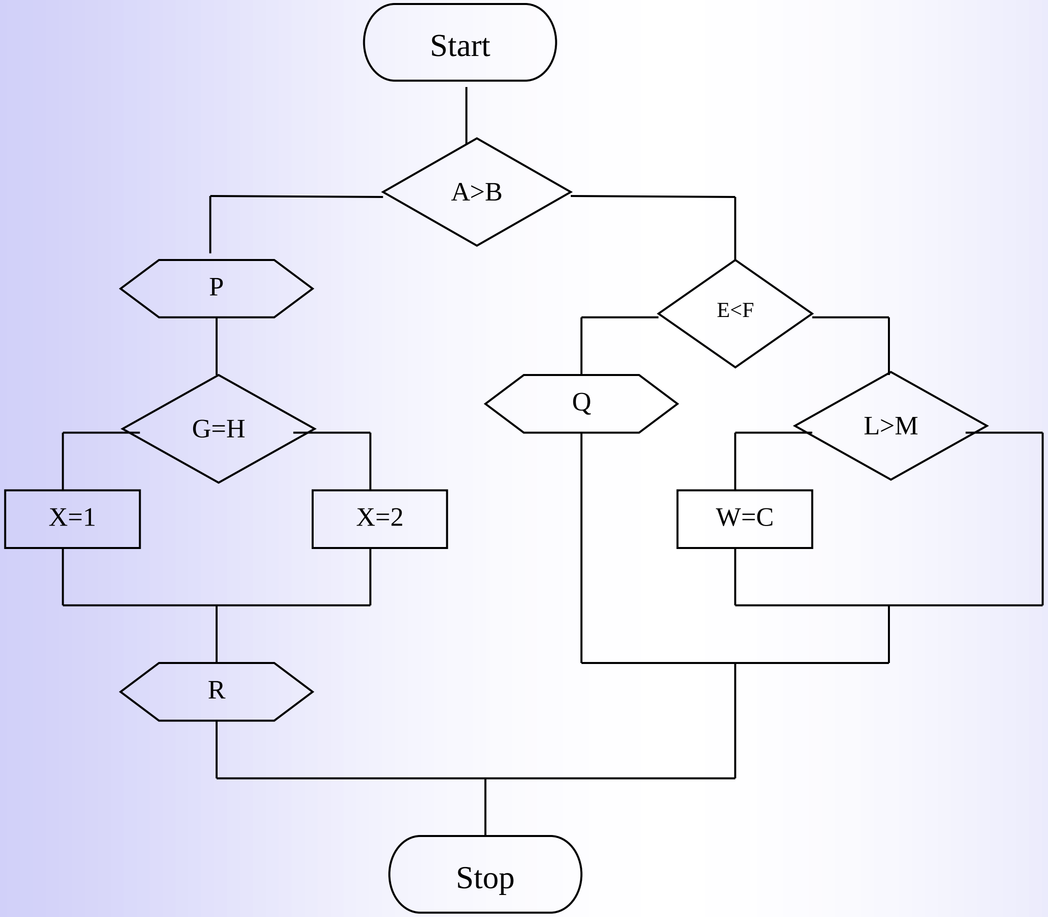
All rights reserved to D.Maymone

Functional IF – Pseudocode

```
If CP
        Call Proc-A
        Call Proc-B
End-If
If CP And CQ And CR
        Call Proc-C
        Call Proc-D
End-If
If CP And (Not CQ)
        Call Proc-E
End-If
If CP And (Not CQ) And CS
        Call Proc-F
End-If
If CP And (Not CQ) And (Not CS)
        Call Proc-G
End-If
If CP And CT
        Call Proc-H
End-If
If (Not CP)  And CU
        Call Proc-I
End-If
If (Not CP)  And (Not CU)
        Call Proc-J
End-If
```

All rights reserved to D.Maymone

Convert to Functional IF (Ex-1)

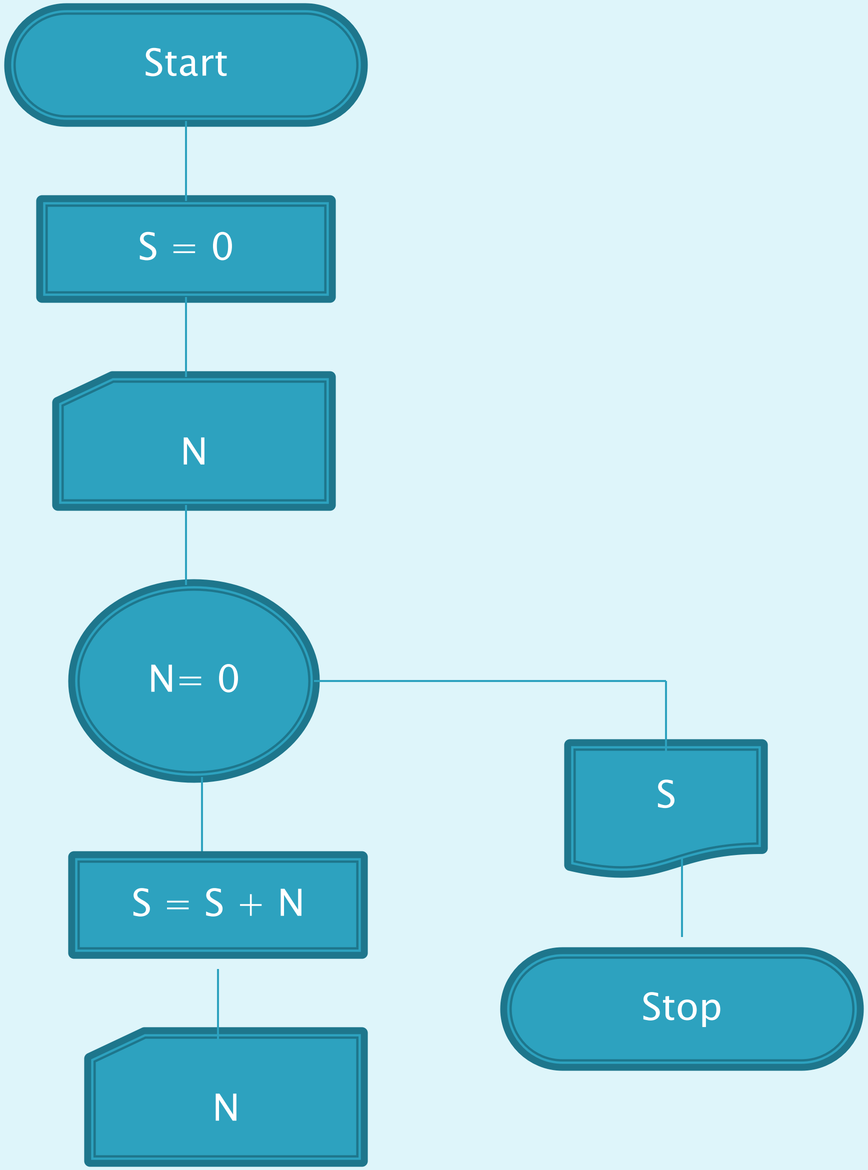


Functional IF הסבר ל-

```
IF A > B Then
    PERFORM P
    IF G = H
       X = 1
    ELSE
       X = 2
    END-IF
    PERFORM R
Else
    IF E = F
       Call Q
    Else
       IF L > M
          W = C
       END-IF
    END-IF
End-IF
```

All rights reserved to D.Maymone

Tree Charts

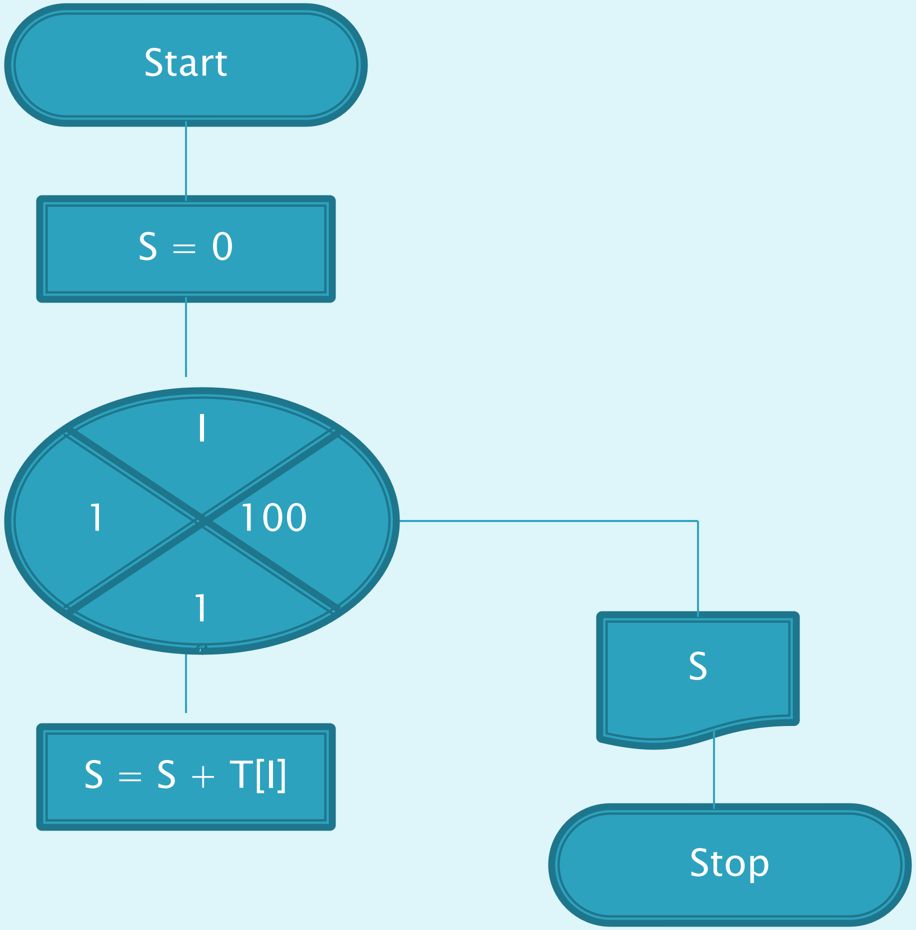


All rights reserved to D.Maymone

Array Sum

Pseudocode:

```
S = 0
For I = 1 To 100
   S = S + T[I]
Next
```

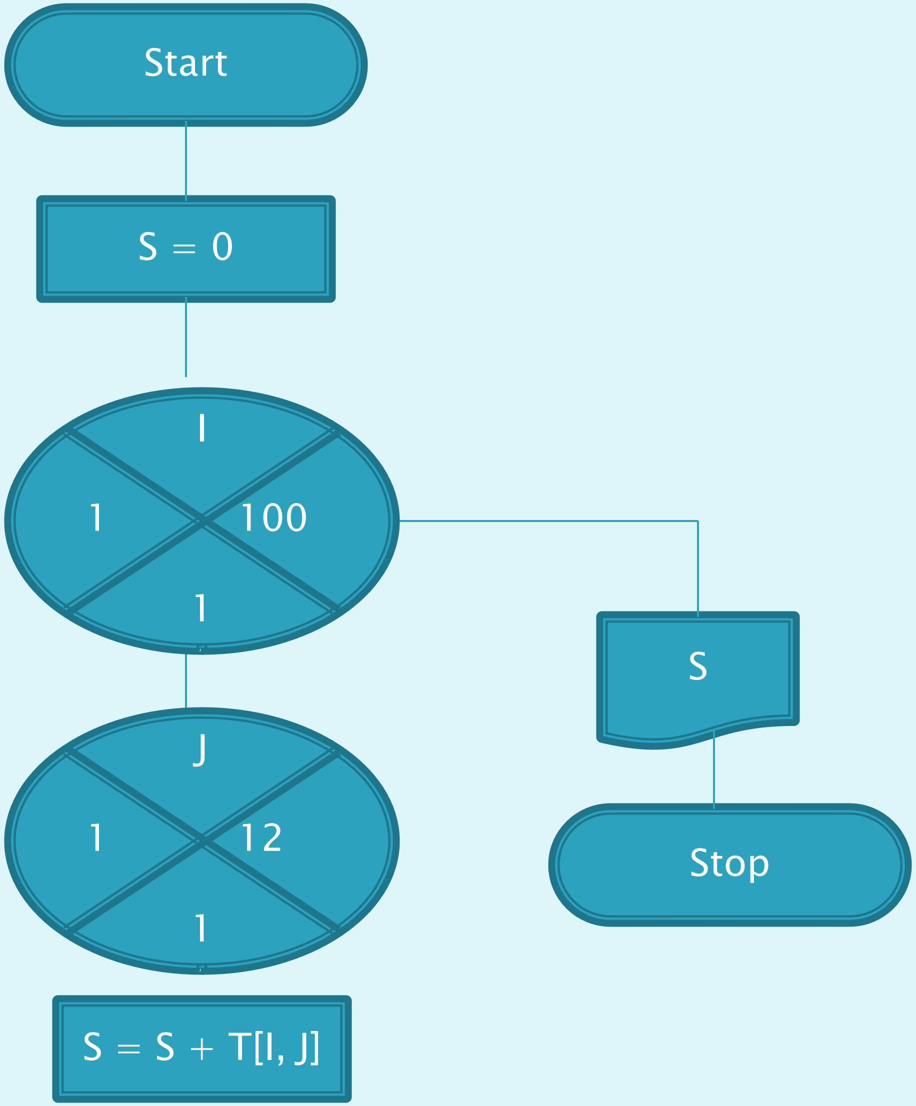


All rights reserved to D.Maymone

Sum of Two Dimensional Array

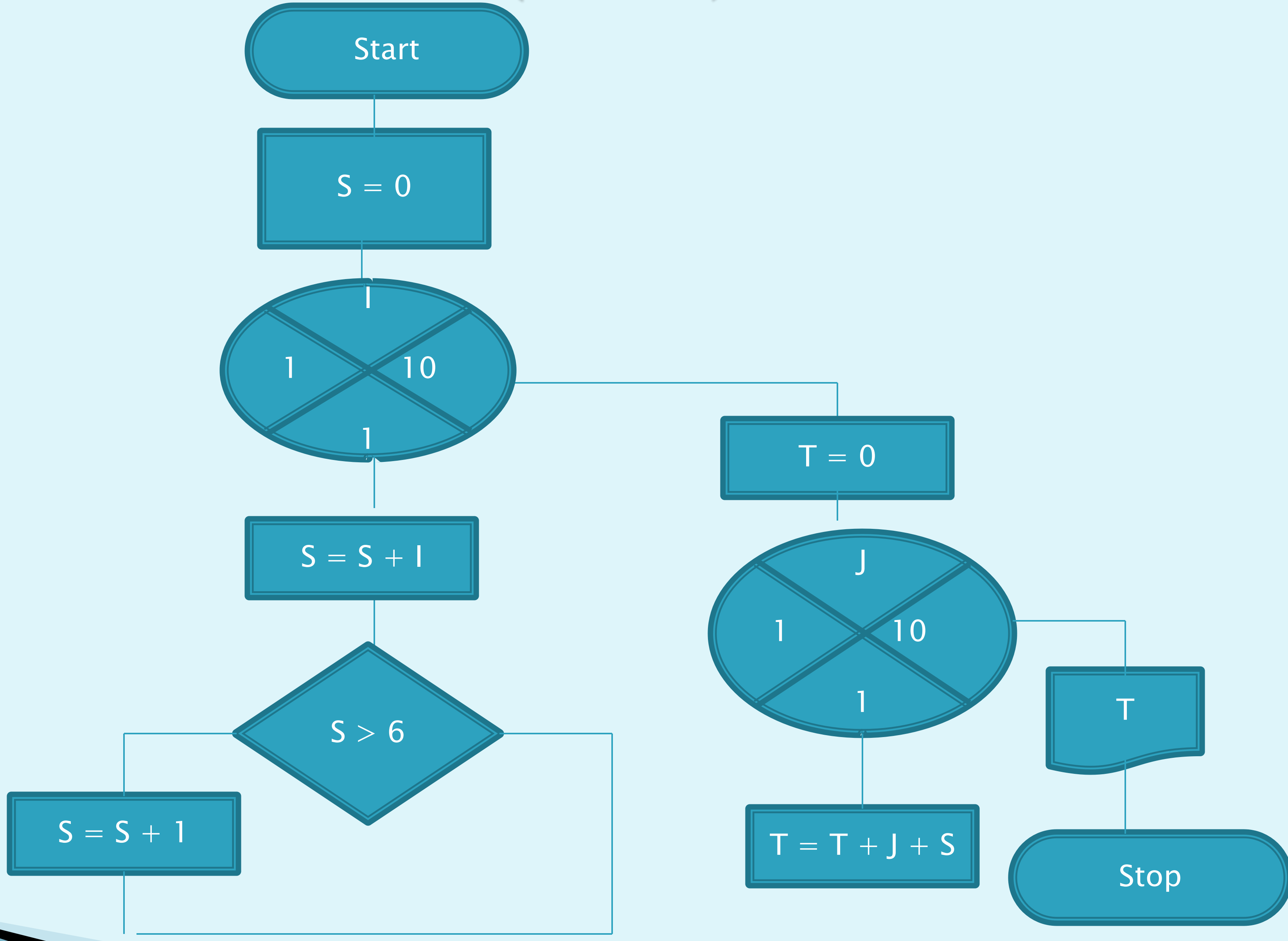
Pseudocode:

```
S = 0
For I = 1 To 100
    For J = 1 To 12
        S = S + T[I, J]
    Next
Next
```



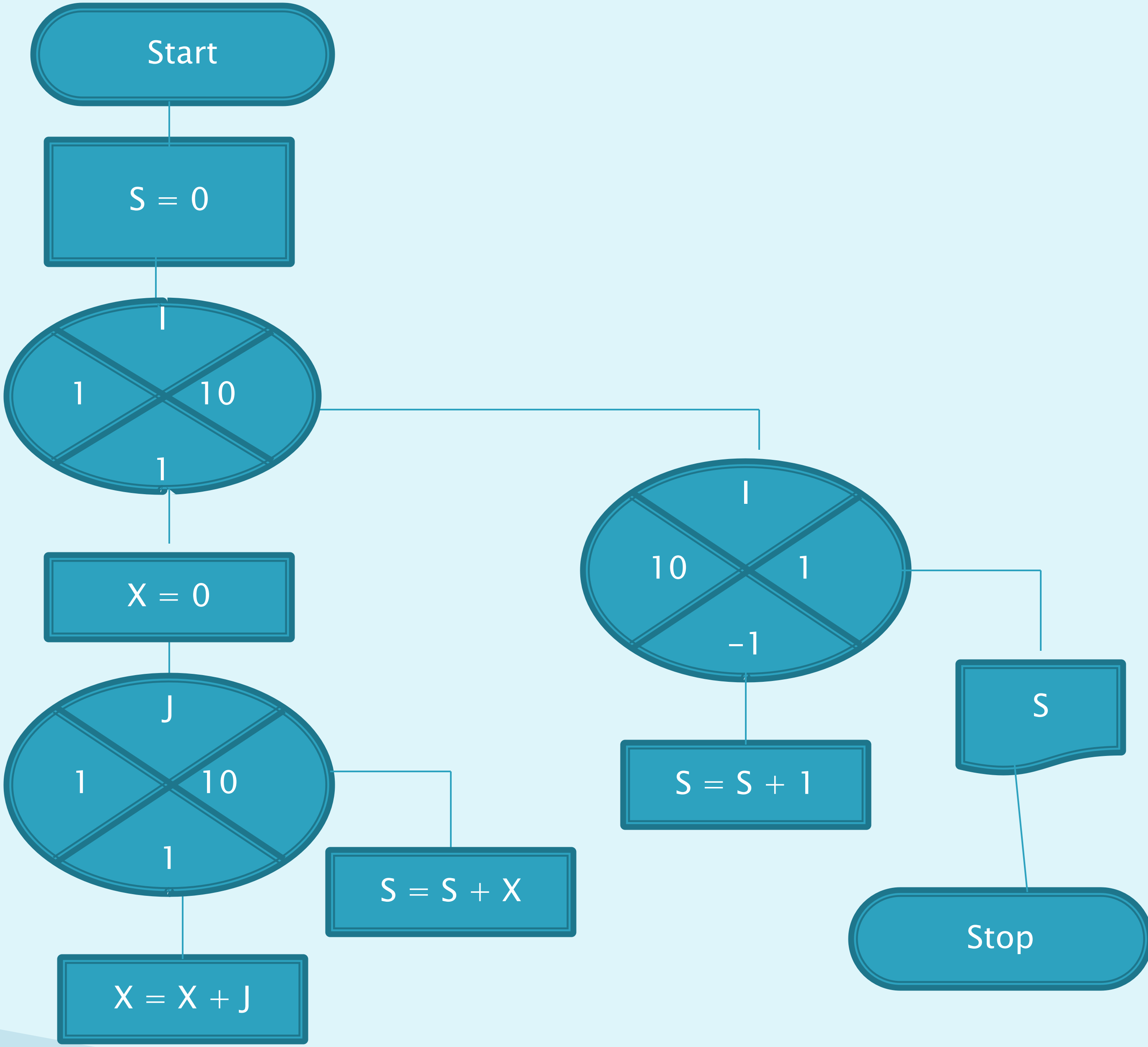
All rights reserved to D.Maymone

Convert to Code (EX-2)



All rights reserved to D.Maymone

Convert to Code (EX-3)



All rights reserved to D.Maymone

שימוש בתאריכים

1. קרא שני תאריכים וחשב את ההפרש בימים ביניהם

2. קרא תאריך ומספר ימים וחשב את התאריך המתקבל ע"י הוספת הימים לתאריך שבקלט
3. קרא מספר בבינרי והצג אותו בדצימלי
4. קרא מספר באוקטלי והצג אותו בדצימלי
5. קרא מספר בהקסדצימלי אותו בדצימלי

All rights reserved to D.Maymone

Exercise 6

נתון מערך דו מימדי של 100 עובדים ו- 12 משכורות
T[100, 12]

1. חשב את סך הכל המשכורות
2. חשב את סל הכל המשכורות של עובד מספר 17
3. חשב את סך הכל המשכורות מלבד המשכורות של העובדים שקיבלו לפחות משכורת אחת גבוהה מ 1,500

All rights reserved to D.Maymone

פקודות מיוחדות בקובול

- MOVE – Reference Modification
- EVALUATE
- Intrinsic Functions
- Search/Search All
- Occurs Depending On
- פקודת CALL
- העברת פרמטרים לתוכנית הראשית

All rights reserved to D.Maymone

REFERENCE MODIFICATION

- VARIABLE (POSITION:SUBSTRING-LENGTH)

  ranges: 1 - FUNCTION LENGTH(VARIABLE)

- 01 EZ-TEXT PIC X(10) VALUE 'characters'.
- MOVE EZ-TEXT(7:4) TO EZ-TEXT(1:3)
- Result: 'terracters'

- MOVE 'aaa' TO EZ-TEXT(7:4)
- Result: 'terracaaa '

EVALUATE

```
         { Identifier     }
         { Literal        }
EVALUATE { CondExpression } ...
         { ArithExpression}
         { TRUE           }
         { FALSE          }

  {      { { ANY                                                                      }     }                }
  {      { { Condition                                                                }     }                }
  { WHEN { { TRUE                                                                     } ... } StatementBlock } ...
  {      { { FALSE                                                                    }     }                }
  {      { {       { Identifier      } [ { THRU    } { Identifier      } ]            }     }                }
  {      { { [NOT] { Literal         } [ { THROUGH } { Literal         } ]            }     }                }
  {      { {       { ArithExpression } [           { ArithExpression } ]            }     }                }

  [WHEN OTHER StatementBlock]
END - EVALUATE
```

EVALUATE – Format 1

```
EVALUATE TRUE
    WHEN condition-1
        statement-block-1
    WHEN condition-2
        statement-block-2
    WHEN OTHER
        statement-block-3
END-EVALUATE
```

EVALUATE – Format 2

```
EVALUTATE Identifier
    WHEN { IDENTIFIER      } [ (THRU) { IDENTIFIER      } ]
         { LITERAL         } [        { LITERAL         } ]
         { ARITHEXPRESSION } [        { ARITHEXPRESSION } ]
            StatementBlock
    WHEN { IDENTIFIER      } [ (THRU) { IDENTIFIER      } ]
         { LITERAL         } [        { LITERAL         } ]
         { ARITHEXPRESSION } [        { ARITHEXPRESSION } ]
            StatementBlock
    WHEN OTHER
            StatementBlock
END-EVALUATE
```

דוגמה - EVALUATE – Format 2

```
EVALUATE EZ01-RANK
        WHEN 1 THRU 17
                ADD 1000 TO EZ02-SALARY
        WHEN 18
                ADD 3000 TO EZ02-SALARY
        WHEN OTHER
                DISPLAY ’NOT IN RANGE’
END-EVALUATE
```

תרגילים

החלף את ה- IF ב- EVALUATE בקוד הבא

```
IF  A > 5  AND  B > 9
THEN
       PERFORM G-HANDLE
ELSE
       IF  A = 1 OR  A = 3 OR A = -5  OR B >= 2
       THEN
              PERFORM   H-SUM
       ELSE
              IF (A = 4 AND B = 1) OR B = 0
              THEN
                     IF  C  >  50
                     THEN
                            DISPLAY 'YES !'
                     END-IF
              END-IF
       END-IF
END-IF
```

Intrinsic Functions

- מבוא
- סוגי הפונקציות
- סוגי הפרמטרים
- פונקציות מחרוזת
- פונקציות סטטיסטיות
- פונקציות מתימטיות
- פונקציות תאריכים
- תרגילים

Intrinsic FUNCTIONs

- Intrinsic FUNCTIONs הם פונקציות פנימיות של קובול.
- לא מדובר ב- User defined Functions
- הפונקציה מחזירה ערך שניתן לשלבו בביטויים מתאימים

המשך - Intrinsic FUNCTIONs

- ניתן להשתמש בתוצאה של פונקציה כאחד מהאופרנדים של פקודת קובול או כפרמטר לפונקציה אחרת.
- לא ניתן להשתמש בפונקציה כיעד של חישוב.
- צורת השימוש:

FUNCTION <u>Function-name (Arguments)</u>

סוגי הפונקציות

- Alphanumeric
  - לטיפול במחרוזות/טקסט
- Numeric
  - מחזירות ערך עם סימן. אפשר להשתמש בהן רק בביטויים אריתמטיים
- Integer
  - ערכים נומריים ללא V ועם סימן

פרמטרים

- הפרמטרים בתוך הסוגריים יכולים להיות:
  - משתנים (כולל כניסה בטבלה)
  - ביטויים אריתמטיים
  - קבועים
- מספר הפרמטרים יכול להיות:
  - אפס (סוגריים ריקים)
  - אחד (לדוגמה ב- SQRT)
  - יותר מאחד (לדוגמה ב- REM)

סוגי הפרמטרים

- NUMERIC
- ALPHABETIC
- ALPHANUMERIC
- שלם

טבלאות כפרמטר לפונקציה

- כאשר פונקציה מאפשרת ריבוי פרמטרים, ניתן להשתמש ב- ALL כדי לציין את כל הכניסות של הטבלה
- אם בטבלה צוין גם OCCURS...DEPENDING ON אז מספר הכניסות נקבע ע"י המשתנה שצוין ב- DEPENDING ON
- דוגמה:

```
COMPUTE WS10-TOTAL =
FUNCTION SUM (TB01-NUM(ALL))
```

פונקציות מחרוזתיות

פונקציה	פרמטרים	סוג הפונקציה	הערך המוחזר
CHAR	int	Alphanumeric	התו הנמצא במיקום int בטבלת ה- EBCDIC/ASCII
LENGTH	any	Integer	אורך המחרוזת
LOWER-CASE	string	Alphanumeric	הפיכת המחרוזת לאותיות קטנות
NUMVAL	alphanumeric	Numeric	הפיכת הערך שבפרמטר לנומרי
ORD	string	Integer	מיקום התוו בטבלת ה- ASCII/EBCDIC
REVERSE	string	Alphanumeric	המחרוזת בסדר הפוך
UPPER-CASE	string	Alphanumeric	הפיכת המחרוזת לאותיות גדולות

פונקציות מחרוזתיות - דוגמאות

```
MOVE FUNCTION CHAR(FUNCTION ORD('a'))
 TO WS20-CHAR-X
COMPUTE WS30-ORDENAL-ASCII-POSITION-I =
              FUNCTION ORD (FUNCTION CHAR(61))
MOVE FUNCTION LENGTH (WS20-CHAR-X)
 TO CN10-ORIGINAL-VAR-LENGTH
IF FUNCTION LOWERCASE(WS20-CHAR-X)= WS20-CHAR-X
THEN
   DISPLAY 'The value is lowercase'
END-IF
IF FUNCTION UPPERCASE(WS20-CHAR-X)= WS20-CHAR-X
THEN
   DISPLAY 'The value is Uppercase'
END-IF
COMPUTE WS01-NUM-N =
             FUNCTION NUMVAL(FUNCTION REVERSE('12'))
```

פונקציות סטטיסטיות

פונקציה	פרמטרים	סוג	הערך המוחזר
MAX	1 or more, any	Any	הערך המקסימלי
MEAN	1 or more, num	Numeric	ממוצע
MIDRANGE	1 or more, num	Numeric	ממוצע של המינימום והמקסימום
MIN	1 or more, any	Any	ערך מינימלי
ORD-MAX	1 or more, any	Integer	מיקום הערך המקסימלי
ORD-MIN	1 or more, any	Integer	מיקום הערך המינימלי
RANGE	1 or more, int/num	Any	ההפרש בין הערך המקסימלי לערך המינימלי
STANDARD-DEVIATION	1 or more, num	Numeric	סטיית התקן
SUM	1 or more, num	Numeric	סכום הפרמטרים
VARIANCE	1 or more, num	Numeric	שונות

פונקציות סטטיסטיות - דוגמאות

```
COMPUTE WS02-NUM-V99 =
  FUNCTION MAX[MIN](WS01-INTEGER TB01-NUM(ALL 3))

COMPUTE WS01-INTEGER =
  FUNCTION ORD-MAX[ORD-MIN](TB01-NAME(ALL ALL))

COMPUTE WS02-NUM-V99 =
  FUNCTION MEAN[STANDARD-DEVIATION][MEDIAN]
  [MIDRANGE][RANGE][SUM][VARIANCE](TB01-NUM(ALL ALL))
```

פונקציות מתימטיות

פונקציה	פרמטרים	סוג
ACOS	1,num	Numeric
ASIN	1,num	Numeric
ATAN	1,num	Numeric
COS	1,num	Numeric
FACTORIAL	1,int	Integer
INTEGER	1,num	Integer
INTEGER-PART	1,num	Integer
LOG	1,num	Numeric
LOG10	1,num	Numeric
MOD	2,int1 int2	Integer
RANDOM	1,int or none	Numeric
REM	2,num1 num2	Numeric
SIN	1,num	Numeric
SUM	1 or more, int/num	Numeric
SQRT	1,num	Numeric
TAN	1,num	Numeric

מה ההבדל בין INTEGER ו- INTEGER-PART? – יש לחפש ב- HELP

פונקציות מתימטיות - דוגמאות

```
COMPUTE WS04-INT =
  FUNCTION MOD(WS03-NUM-V99   5)

COMPUTE WS02-NUM-V99 =
  FUNCTION REM (WS03-INT1   WS04-INT2)

COMPUTE CN01-FACTORIAL =
  FUNCTION FACTORIAL (WS01-INTEGER)

COMPUTE WS01-INTEGER =
  FUNCTION INTEGER[-PART] (WS02-NUM-V99)

COMPUTE WS01-SUM =
  FUNCTION SUM(TB01-SALARIES(ALL 2))
```

משמעות: סכם את המשכורות של החודש **השני** של **כל** העובדים.

פונקציות תאריכים

- הלוח הגרגוריאני נבחר לשמש את פונקציות התאריכים
- התאריך ההתחלתי הוא יום שני, 1 ינואר 1601. וזהו יום מספר 1.
- כל פונקציות התאריכים מתרגמות את התאריך למספר הסידורי שלו בלוח הגרגוריאני.

פונקציות תאריכים

פונקציה	פרמטרים	סוג	הערך המוחזר
CURRENT-DATE	None	Alphanumeric	תאריך נוכחי YYYYMMDDhhmmsscccccc
DATE-OF-INTEGER	1,int	Integer	תרגום מספר לתאריך בצורה YYYYMMDD
DAY-OF-INTEGER	1,int	Integer	תרגום מספר לתאריך בצורה YYYYDDD
INTEGER-OF-DATE	1,int	Integer	תרגום תאריך בצורה YYYYMMDD למספר
INTEGER-OF-DAY	1,int	Integer	תרגום תאריך בצורה YYYYDDD למספר

פונקציות תאריכים - דוגמאות

חישוב גיל בימים

```
03 WS01-CURRENT-DATE.
        05 WS01-CURRENT-DATE-9    PIC 9(8).
03 WS02-BIRTH-DATE                PIC 9(8) VALUE 19451203.
03 WS03-AGE-IN-DAYS               PIC 9(5).
MOVE FUNCTION CURRENT-DATE TO WS01-CURRENT-DATE
COMPUTE WS03-AGE-IN-DAYS =
    FUNCTION INTEGER-OF-DATE(WS01-CURRENT-DATE) –
                         FUNCTION INTEGER-OF-DATE(WS02-BIRTH-DATE)
```

פונקציות תאריכים - המשך

איך לחשב את היום בשבוע?

השתמש בשארית החלוקה ב- 7

```
COMPUTE WS10-DATE-DAY-OF-WEEK =
  FUNCTION REM (FUNCTION INTEGER-OF-
  DATE (WS11-DATE-YYYYMMDD)   7)
```

הפונקציה תחזיר: 0 עבור יום ראשון, 1 עבור יום שני...

תרגיל 11.1

- נתונים 4 ציונים , כתוב תוכנית להדפסת המכסימום
- יש להשתמש בערכים כדלקמן

A=68

B=75

C=95

D=88

SEARCH Syntax

$$\underline{\textit{SEARCH}} \quad \text{TableName} \left[\underline{\text{VARYING}} \begin{Bmatrix} \text{Indentifier} \\ \text{IndexName} \end{Bmatrix} \right]$$

$$\left[\text{AT } \underline{\text{END}} \text{ StatementBlock} \right]$$

$$\left\{ \underline{\text{WHEN}} \text{ Condition} \begin{Bmatrix} \text{Statementblock} \\ \underline{\text{NEXT SENTENCE}} \end{Bmatrix} \right\} \ldots$$

$$\underline{\textit{END} - \textit{SEARCH}}$$

$$\underline{\textbf{OCCUR}}\textbf{S} \quad \textbf{TableSize} \quad \textbf{TIMES}$$

$$\left[\underline{\textbf{INDEXED}} \textbf{ BY } \{\textbf{IndexName}\} \ldots \right]$$

משפט ה- SET

```
SET IndexName/Identifier … TO

    IndexName/Identifier/Integer


SET IndexName… UP/DOWN

    BY Identifier/Integer
```

SEARCH - הערות

- שם המשתנה המופיע בפקודת ה- SEARCH חייב לכלול בהגדרה שלו את פסקת ה- OCCURS
- שימוש ב- VARYING בפקודת SEARCH הוא לא סטנדרטי ולכן כמעט שאין שימוש בו
- כנ"ל לגבי NEXT SENTENCE (הפיקוח עובר לפקודה אחרי ה- SEARCH)
- ניתן להגדיר מספר INDEXED BY בטבלה אחת אבל פקודת ה- SEARCH משתמשת בראשון בלבד (ברירת המחדל)
- ב- VARYING ניתן לקבוע באיזה INDEXED BY להשתמש
- האינדקס יכול להיות תוצאה של חישוב
- החישוב מבוצע ע"י הפקודה : SET… UP/DOWN BY

תחביר ה-SEARCH ALL

```
SEARCH ALL TableName     [AT END StatementBlock]

      WHEN { ElementIdentifier { IS EQUAL TO } { Identifier       }  }
           {                   { IS =        } { Literal          }  }
           {                                   { ArithExpression  }  }
           { ConditionName                                           }

    [ AND { ElementIdentifier { IS EQUAL TO } { Identifier       }  } ]
    [     {                   { IS =        } { Literal          }  } ]
    [     {                                   { ArithExpression  }  } ]
    [     { ConditionName                                           } ]

    { StatementBlock }
    { NEXT SENTENCE  }

END – SEARCH


          OCCURS TableSize TIMES

             [ { ASCENDING  } KEY IS {ElementIdentifier} … ] …
             [ { DESCENDING }                             ]

             [INDEXED BY {IndexName}…]
```

SEARCH ALL - הערות

- ה- SEARCH ALL פועל רק על טבלאות ממוינות (ASCENDING/DESCENDING)
- פקודת ה- SEARCH ALL לא בודקת אם הטבלה ממוינת. (אחריות התוכניתן)
- בפקודת ה- SEARCH ALL יש שימוש רק בתנאי שוויון אחד
- ה- SEARCH ALL מתחיל את החיפוש תמיד מתחילת הטבלה (אינו מתייחס לערך של האינדקס האישי).

DEPENDING ON

```
01   TB-TABLES.
     03 TB01-LETTERS.
        05 TB01-LETTER    PIC X
                          OCCURS 1 TO 26
                          INDEXED BY TB01-I
                          DEPENDING ON CN01-TABLE-SIZE.
```

- ניתן להשתמש בפסקת ה- DEPENDING ON כדי לצמצם את טווח הטבלה
- ה- DEPENDING ON לא משנה את גודל הזכרון בשימוש הטבלה

CALL תחביר

CALL ProgramNameIdentifier | ProgramNameLiteral

[USING [BY REFERENCE | BY CONTENT] Param1

[BY REFERENCE | BY CONTENT] Param2…]

…….

END-CALL

LINKAGE SECTION

- מגדיר את מבנה הפרמטרים/ארגומנטים
- אין להגדיר VALUE בתוך ה- LINKAGE SECTION מלבד ברמה 88
- מומלץ שסדר הופעת הפרמטרים יהיה זהה ל- USING
- UPSI – אין להשתמש בתוכנית חדשות
- מומלץ להוסיף תמיד פרמטר אחרון שיכיל את הסטטוס (Return Code)

CALL BY VALUE

- בד"כ בשימוש אם קוראים לתוכניות כתובות בשפת תכנות אחרת (בד"כ שפת C)
- אפשר לקרוא BY VALUE גם לתוכניות בשפת COBOL. במקרה זה יש לרשום גם ב- PROCEDURE DIVISION של התוכנית הנקראת את האופציה BY VALUE עבור אותם פרמטרים

CALL RETURNING

- ה- RETURNING משמש להחזרת משתנה עם ערך דרוש
- במקרה ויש שימוש בפקודת CALL באופציה RETURNING יש לציין באותה צורה את ה- RETURNING גם ב- PROCEDURE DIVISION של התוכנית הנקראת.

CANCEL

- CANCEL identifier/literal
- התוכנית "יורדת" מהזכרון.
- בפעם הבאה שהתוכנית תיקרא היא תיטען במצבה הראשוני (ה- VALUE ב- -WORKING STORAGE יהיה תקף)

INITIAL Mode

- PROGRAM-ID. progtramName IS INITIAL.
- בשימוש כאשר התוכנית נקראת מתוכנית אחרת
- לפני שהתוכנית מתחילה לפעול מערכת ההפעלה מחזירה אותה למצבה ההתחלתי.
- המשמעות העיקרית היא שכל ה- Values שהוגדרו ב- Working-Storage מופעלים שוב.
- אין להשתמש באופציה זו במתף.
- אפשר להשתמש ב- Local Storage

INSPECT Syntax

Format 1

```
INSPECT SourceStr$i TALLYING

                         { CHARACTERS [ {BEFORE} INITIAL Delim$il ] ...                              }
                         {              {AFTER }                                                     }
   Counter#i FOR         {                                                                           } ...
                         { {ALL    } { Compare$il [ {BEFORE} INITIAL Delim$il ] ... } ...             }
                         { {LEADING} {            [ {AFTER } ]                      }                 }
```

```
INSPECT WS01-FULL-NAME
        TALLYING CN01-SPACES
         FOR LEADING SPACES.

INSPECT WS01-SOURCE-LINE
        TALLYING CN01-NO-OF-ES
        FOR ALL 'e' AFTER  INITIAL 'start'
                    BEFORE INITIAL 'end'.
```

INSPECT Syntax

Format 2

INSPECT SourceStr$i **REPLACING**

{ **CHARACTERS** BY ReplaceChar$il [{ **BEFORE** / **AFTER** } INITIAL Delim$il] …

{ **ALL** / **LEADING** / **FIRST** } { Compare$il BY Replace$il [{ **BEFORE** / **AFTER** } INITIAL Delim$il] … } … } …

```
PERFORM VARYING I
        FROM 1 BY 1
        UNTIL I > 10
  INSPECT WS01-TEXT-LINE
            REPLACING TB01-SWEAR-WORD(I) BY '*#@!'
END-PERFORM
```

INSPECT Syntax
Format 3

INSPECT SourceStr$i **TALLYING**

Counter#i **FOR**
{ **CHARACTERS** [{ **BEFORE** / **AFTER** } INITIAL Delim$il] …
{ **ALL** / **LEADING** } { De lim $il [{ **BEFORE** / **AFTER** } INITIAL Delim$il] … } … } …

REPLACING
{ **CHARACTERS** BY ReplaceChar$il [{ **BEFORE** / **AFTER** } INITIAL Delim$il] …
{ **ALL** / **LEADING** / **FIRST** } { Compare$il BY Replace$il [{ **BEFORE** / **AFTER** } INITIAL Delim$il] … } … } …

INSPECT Syntax

Format 4

INSPECT SourceStr$i CONVERTING

Compare$il TO Convert$il

[{ **BEFORE** / **AFTER** } **INITIAL Delim$il**] ...

```
INSPECT WS01-TEXT CONVERTING
    '0123456789' TO '5298317046'
    AFTER  INITIAL 'codeon'
    BEFORE INITIAL 'codeoff'.
```

STRING Syntax

$$\textbf{STRING}\ \left\{ \textbf{Source\$il}\ \ldots\ \underline{\textbf{DELIMITED}}\ \textbf{BY} \begin{Bmatrix} \textbf{Delim\$il} \\ \underline{\textbf{SIZE}} \end{Bmatrix} \right\} \ldots$$

$$\underline{\textbf{INTO}}\ \textbf{Dest\$i}$$

$$[\textbf{WITH}\ \underline{\textbf{POINTER}}\ \textbf{Pointer\#i}]$$

$$[\textbf{ON}\ \underline{\textbf{OVERFLOW}}\ \textbf{StatementBlock}]$$

$$[\underline{\textbf{NOT}}\ \textbf{ON}\ \underline{\textbf{OVERFLOW}}\ \textbf{StatementBlock}]$$

$$[\underline{\textbf{END - STRING}}]$$

```
STRING   WS01-SOURCE-1
           WS02-SOURCE-2
           '10'          DELIMITED BY SIZE
         INTO WS03-DEST-STRING
END-STRING
STRING   WS01-SOURCE-1 DELIMITED BY SIZE
         WS02-SOURCE-2 DELIMITED BY SPACES
         WS03-SOURCE-3 DELIMITED BY 'Frogs'
         INTO WS04-DEST
          WITH POINTER IX01-POINTER
END-STRING
```

UNSTRING Syntax

UNSTRING SourceStr$i
    [DELIMITED BY [ALL] Delim$il [OR [ALL] Delim$il] ...]
    INTO {DestStr$i [DELIMITER IN HoldDelim$i]
            [COUNT IN CharCounter# i]} ...
    [WITH POINTER Pointer# i]
    [TALLYING IN DestCounter# i]
    [ON OVERFLOW StatementBlock]
    [NOT ON OVERFLOW StatementBlock]
    [END - UNSTRING]

```
UNSTRING WS01-Name
       DELIMITED BY ALL SPACES
       INTO WS01-FIRST-NAME
                 WS01-SECOND-NAME
                 WS01-SURNAME
END-UNSTRING
```

```
UNSTRING WS01-ADDRESS DELIMITED BY ','
   INTO TB01-ADDRESS-LINE(1)
             TB01-ADDRESS-LINE(2)
             TB01-ADDRESS-LINE(3)
             TB01-ADDRESS-LINE(4)
             TB01-ADDRESS-LINE(5)
   TALLYING IN CN01-LINES-USED
END-UNSTRING
```

העברת פרמטרים לתוכנית הראשית

- שימוש ב- UPSI
- שימוש ב- PARM

שימוש ב- UPSI

- CONFIGURATION SECTION.
  - SPECIAL-NAMES.
    - UPSI-0 ON STATUS IS FIRST-UPSI
    - UPSI-1 ON STATUS IS SECOND-UPSI

- IF FIRST-UPSI
  - ….
- END-IF
- IF SECOND-UPSI
  - ….
- END-IF
- // EXEC PGM=TESTUPSI,PARM='/UPSI(10100000)

שימוש ב- PARM

```
//   EXEC PGM=TESTPARM,PARM= 'HELLO, WORLD'

LINKAGE SECTION.
01   LS01.
        03   LS01-LENGTH     PIC S9(4) COMP.
        03   LS01-CONTENT    PIC X(100).
```

טבלאות החלטה

- דוגמה: תוספת לעובד
- תהליך הבניה
- טבלת המשתנים
- טבלת החלטה
- תרגילים

All rights reserved to D.Maymone

תוספת לעובד

- עובדת עם יותר מ- 15 שנות ותק תקבל תוספת של 250
- עובד שגילו עד 45 עם יותר מ-3 ילדים ויותר מ- 15 שנות ותק יקבל תוספת של 400
- עובדת עם יותר מ-3 ילדים תקבל תוספת של 500
- עובד שגילו יותר מ-45 וותק של עד 15 שנה יקבל תוספת של 200

All rights reserved to D.Maymone

תהליך הבניה

- הגדרת משתנים
- קביעת מספר האפשרויות לכל משתנה
- חישוב הכופל עבור כל משתנה
- חישוב מספר האפשרויות הכללי = N
- בניית טבלה של N שורות
- מילוי ערכים לכל המשתנים עם כל הצירופים
- מילוי ההחלטה עבור כל צירוף

All rights reserved to D.Maymone

שאלה

- האם, לפי החוקים האלו, יכול לקבל עובד כלשהו יותר מתוספת אחת?
- מי הם העובדים שלא יקבלו אף תוספת?

All rights reserved to D.Maymone

צורת הקלט

- מספר זהות
- מין : 0 – נקבה 1-זכר
- שנת תחילת עבודה – 4 ספרות
- שנת לידה – 4 ספרות
- מספר ילדים – 2 ספרות

All rights reserved to D.Maymone

טבלת המשתנים

מספר האפשרויות	ערכים	משתנים	שם קצר	כופל
2	0 - נקבה 1 - זכר	מין	G	1
2	0 - עד 15 1 - יותר מ- 15	ותק	Sk	2
2	0 - עד 3 ילדים 1 - יותר מ-3 ילדים	מספר ילדים	C	4
2	0 - עד 45 1 - יותר מ-45	גיל	A	8

- כופל של 1 = 1
- כופל I = כופל של I – 1 * מספר אפשרויות של I – 1

מספר הכללים = מכפלת האפשרויות

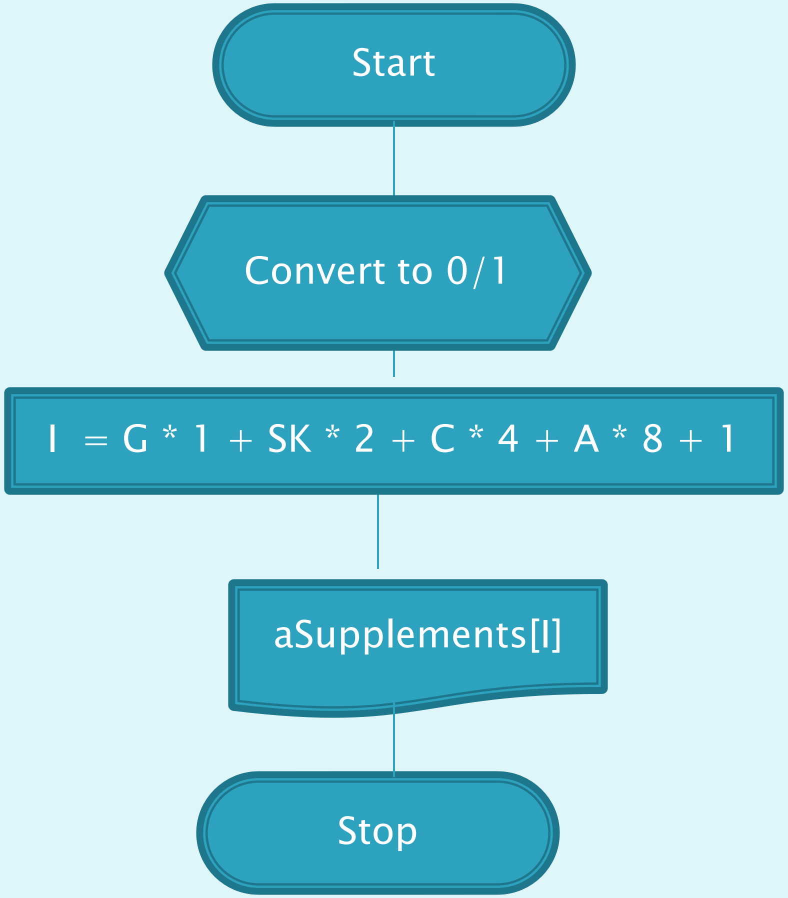
All rights reserved to D.Maymone

טבלת החלטה

כופל משתנים	8	4	2	1	תוספת
	A	C	Sk	G	
1	0	0	0	0	
2	0	0	0	1	
3	0	0	1	0	250
4	0	0	1	1	
5	0	1	0	0	500
6	0	1	0	1	
7	0	1	1	0	**<u>250 ? 500</u>**
8	0	1	1	1	400
9	1	0	0	0	
10	1	0	0	1	
11	1	0	1	0	250
12	1	0	1	1	200
13	1	1	0	0	500
14	1	1	0	1	
15	1	1	1	0	**<u>250 ? 500</u>**
16	1	1	1	1	200

All rights reserved to D.Maymone

Convert to Activity Diagram

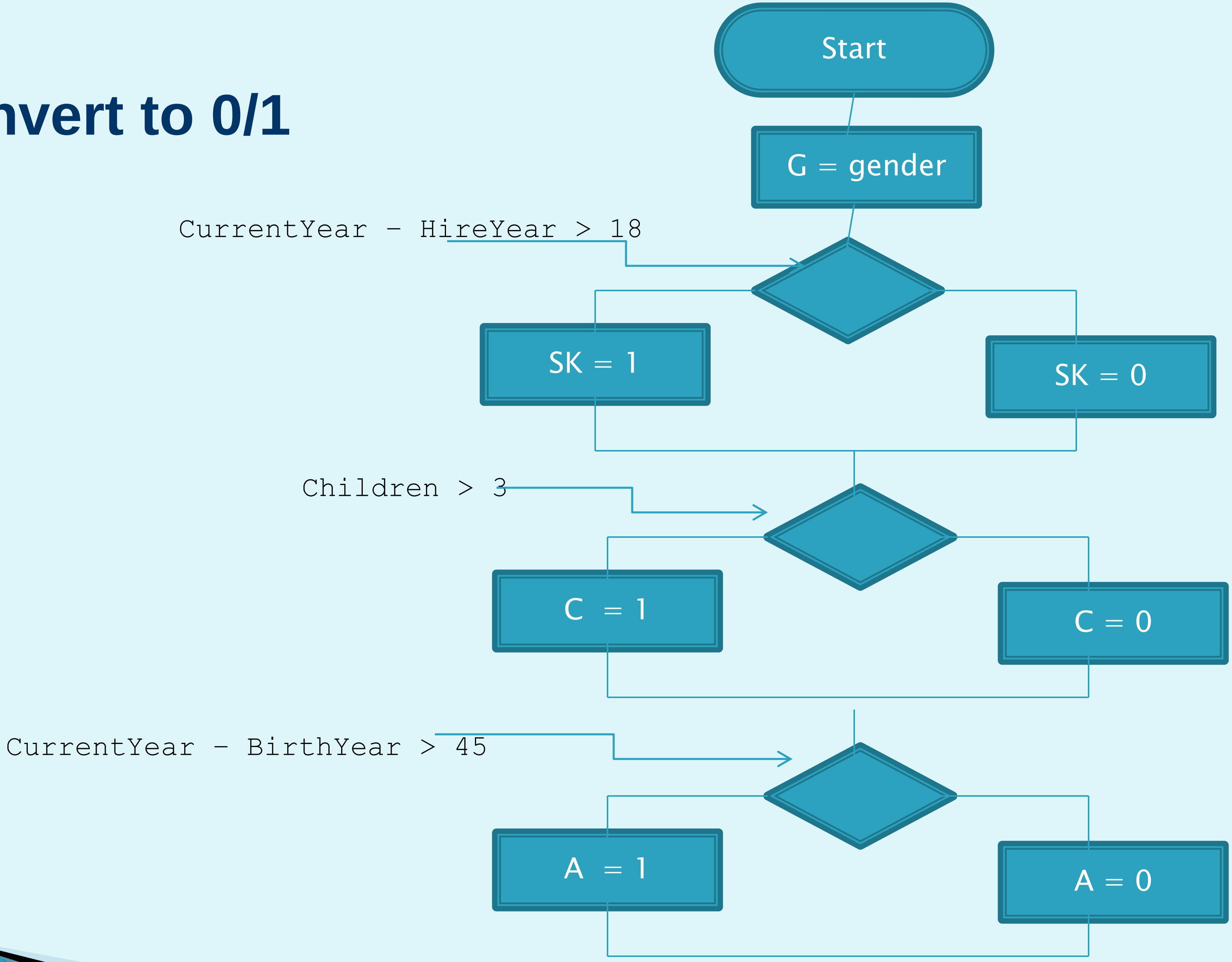


Supplements Table
0
0
0
250
0
500
0
500
400
0
0
250
200
500
0
500
200

כדאי להוסיף את טבלת ההחלטה כתיעוד בתוכנית

All rights reserved to D.Maymone

Convert to 0/1



All rights reserved to D.Maymone

תרגילים

<u>בעיית ביטוח</u>

- אם המין זכר והגיל נמוך מ- 24, יש לשלם פרמיה רגילה של 6% ופרמיה נוספת של 4%.
- אם המין נקבה והגיל נמוך מ- 24, יש לשלם פרמיה רגילה של 6% ופרמיה נוספת של 3%
- אם הגיל נמוך מ- 60 ומספר התאונות = 0, יש לשלם פרמיה רגילה של 4% והנחה של 1%
- אם הגיל בין 24-60 ומספר התאונות לפלות 1, יש לשלם פרמיה רגילה של 4% ופרמיה נוספת של 3%
- אם הגיל גדול מ- 60, ומספר התאונות = 0, יש לשלם פרמיה רגילה של 7%
- אם הגיל גבוה מ- 60 ומספר התאונות לפחות 1, יש לשלם פרמיה רגילה של 7% ופרמיה נוספת של 4%

All rights reserved to D.Maymone

תרגילים

<u>הלוואה ללקוח</u>

- אם הלקוח רוצה הלוואה של עד 10,000 פעל לפי תקנון 20

- אם הלקוח רוצה הלוואה גבוהה מ- 10,000, משלם את חובותיו בזמן ויש לו עסקאות כספיות גדולות, אשר את ההלוואה

- אם הלקוח רוצה הלוואה גבוהה מ- 10,000, לא משלם את חובותיו בזמן ויש לו עסקאות כספיות גדולות, בדוק עם המנהל

- אם הלקוח רוצה הלוואה גבוהה מ- 10,000, לא משלם את חובותיו בזמן ואין לו עסקאות כספיות גדולות, דחה ההלוואה

<u>קלט</u>

סכום ההלוואה

קוד עסקאות כספיות: 0 – קטנות, 1 - גדולות

קוד תשלום חובות: 1 – בזמן, 0 – לא בזמן

All rights reserved to D.Maymone

טבלת המשתנים

מספר האפשרויות	משתנים	שם קצר	ערכים	כופל

All rights reserved to D.Maymone

טבלת ההחלטה

כופל משתנים					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

All rights reserved to D.Maymone

חישוב משכורת לעובד

- אם העובד רווק בצע "טיפול מיוחד"
- אם העובד גרוש עם לפחות ילד אחד בצע "תוספת גירושין"
- אם העובדת אלמנה בצע "תוספת אלמנה"
- אם העובד נשוי ויש לו עד 2 ילדים בצע "תוספת חלקית"
- אם העובד נשוי ויש לו 3 ילדים או יותר בצע "תוספת מיוחדת"
- עבור כל צירוף אחר בצע "ללא תוספת"

All rights reserved to D.Maymone

טבלת המשתנים

מספר האפשרויות	משתנים	שם קצר	ערכים	כופל

All rights reserved to D.Maymone

טבלת ההחלטה

כופל משתנים					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

All rights reserved to D.Maymone

החזר ביטוח רפואי

- בביקור אצל רופא
  - מטופל בגיל פנסיה זכאי ל- 80% החזר.
  - מטופל עד גיל 18 זכאי ל- 100% החזר
  - מטופל אחר זכאי ל- 50% החזר אם שילם לפחות תשלום אחד ב-3 החודשים האחרונים, ל- 70% אם הוא שילם את 3 התשלומים האחרונים. מטופל שלא שילם אף תשלום ב-3 החודשים האחרונים לא זכאי להחזר.
- בביקור בבית חולים
  - מטופל בגיל פנסיה זכאי ל- 90% החזר.
  - מטופל אחר זכאי ל- 70% החזר אם יש לו לפחות תשלום אחד ב- 3 החודשים האחרונים ול- 50% אחרת
- בביצוע בדיקה במעבדה
  - מטופל בגיל פנסיה זכאי ל- 70% החזר.
  - מטופל אחר זכאי ל- 50% החזר אם יש לו לפחות תשלום אחד ב- 3 החודשים האחרונים ול- 30% אחרת

All rights reserved to D.Maymone

תרגילים

<u>בעיית האסירים</u>

- נתונים 3 אסירים : A, B, C, כאשר C עיוור.
- לסוסר 3 כובעים אדומים ו- 2 כובעים לבנים
- הסוהר החליט להשתעשע עם האסירים בצורה הבאה:-
  - על ראשו של כל אסיר יונח כובע כאשר עיני שלושתם קשורות
  - על פי הסדר יוסר הקשר מעל עיניהם ועל פי הכובעים של שני הנותרים, רשאי האסיר לנחש מהו צבע כובעו
  - אסיר שינחש נכון ייצא לחופשי. אסיר שיטעה, יגרום לכך ששלושתם ייכנסו לצינוק לחצי שנה
- A אמר : אני לא יודע
- B אמר : אני לא יודע
- C למרות היותו עיוור ידע את התשובה הנכונה!
- מהי?

All rights reserved to D.Maymone

תרגיל – בדיקות חורף

- אם גיל המכונית מעל 15, נפח המנוע מעל 1800 ומצבה הטכני טוב, יש לבצע בדיקת חורף בלבד.
- אם גיל המכונית מעל 15 ומצבה הטכני גרוע, יש להשבית את המכונית.
- אם גיל המכונית בין 10 ל-15 שנה, נפח המנוע מעל 1800 ומצבה הטכני גרוע, יש לבצע בדיקה כללית ובדיקת חורף.
- אם גיל המכונית קטן מ- 10, ומצבה הטכני גרוע, יש לבצע בדיקה כללית.
- בכל שאר המקרים, יש להדפיס תווית "כשרה לתנועה.

נתון קובץ תנועות במבנה הבא:

- זיהוי מכונית : XX-XXX-XX
- שנת ייצור YYYY
- נפח מנוע: NNNN
- מצב טכני: 0- גרוע, 1-טוב

1. יש לנתח את הבעיה
2. לכתוב תוכנית קובול שתקרא את קובץ התנועות ותציג עבור כל מכונית את הבדיקה שיש לבצע

All rights reserved to D.Maymone

יתרונות השימוש בטבלאות החלטה

- מצמצם זמן תכנון ותכנות
- לוגיקה ברורה ושלמה
- תרגום מהיר לשפת תכנות
- מראה בבירור מהלכים המסובכים מידי עבור תרשימי זרימה
- קשר מעולה בין הלקוח, מנתח המערכות, מעצב ותוכניתן
- תעוד מעולה בגוף התוכנית
- נוח להכנסת שינויים
- מאפשר הכנה יעילה של נתוני ניסוי

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

StateChart Diagram

- מבוא
- דוגמאות של מצבים
- ציון ה- State
- Transition
- בדיקת המודל
- תרגיל

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

מבוא ל- State Diagram

- מציג את המצבים השונים של אובייקט
- איך אירועים שונים משפיעים על המצבים
- נחוץ עבור ה- Classes בהם יש מצבים שונים וב- Classes עם Methods מורכבים
- אובייקט מחליף את מצבו כאשר מתרחש ארוע
- דוגמאות

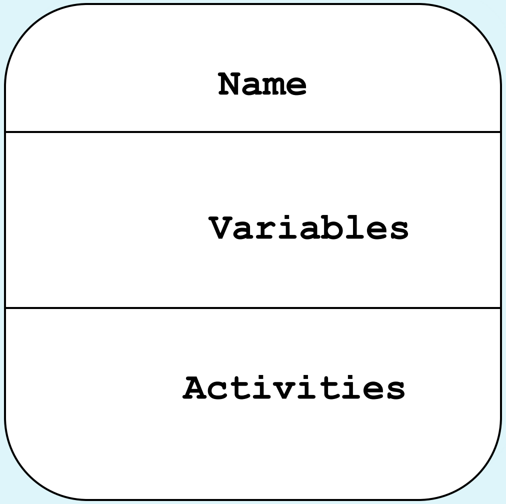
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

דוגמאות של מצבים

- חשבונית (אובייקט) שולמה(מצב)
- המנוע (אובייקט) עובד (מצב)
- המכונית (אובייקט) בנסיעה (מצב)
- הספר (אובייקט) בהשאלה (מצב)
- הספר (אובייקט) בהמתנה (מצב)

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

ציון ה- State



או

- Name : שם המצב (שולמה, בהמתנה, בנסיעה)
- Variables : משתנים המכילים את המצב (סטטוס) או נחוצים להמשך הפעילות
- Activities

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במ[illegible]ם בע"מ

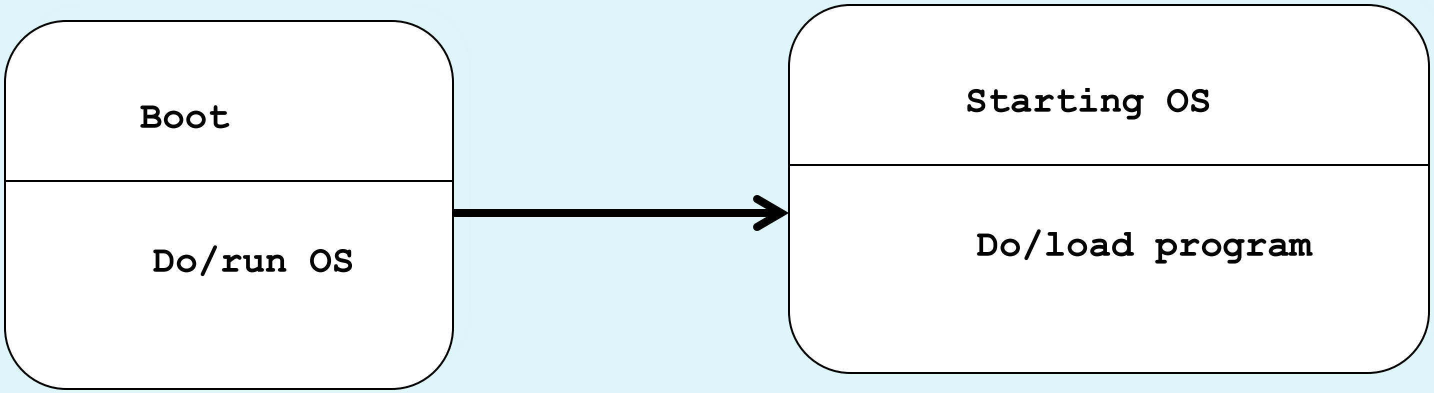
Activities

- Entry : פעולה בכניסה למצב
  - Entry/select login
- Exit : פעולה ביציאה מהמצב
  - exit/login(username)
- Do : פעולה המבוצעת במשך הזמן בו האובייקט נמצא במצב.
  - בהמתנה
  - מחשב (calculating)
  - עדכון סטטוס

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

State Transition

- לפעמים אין צורך בציון מיוחד



דוגמה עם ציון ארוע

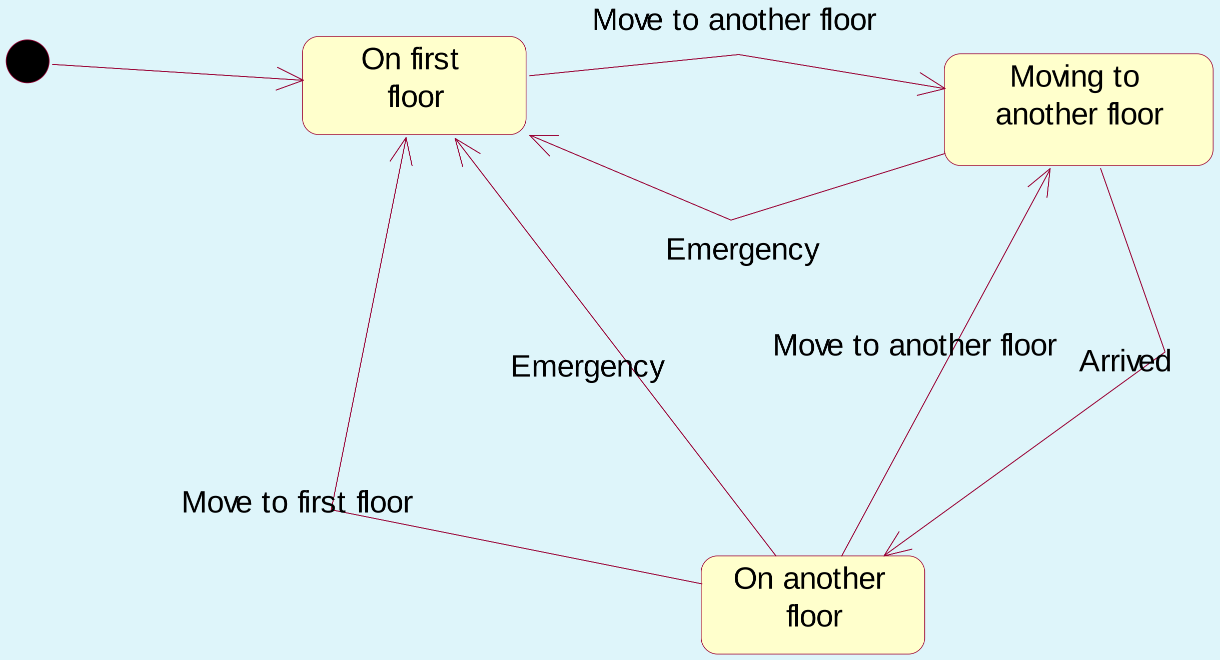
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשוב בע"מ

דוגמה ל- State Diagram
בעיית המעלית

- המעלית נמצאת בהתחלה בקומת קרקע
- בקומת קרקע ניתן לבקש להגיע לקומה כלשהי בין 1 ל- N
- בזמן התנועה המעלית יכולה לקבל קריאת חירום ואז היא חייבת להגיע מיד לקומת הקרקע
- כשהמעלית נמצאת בקומה כלשהי (1 עד N) היא יכולה לקבל פקודה לעלות או לרדת לקומה אחרת (כולל קרקע)
- בכל קומה שהיא מלבד קומת הקרקע אם לא התקבלה פקודה כלשהי במשך 1 דקה המעלית יורדת אוטומטית לקומת קרקע

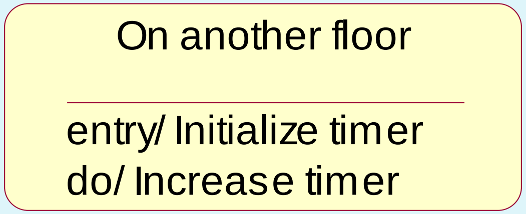
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשבים בע"מ

The Elevator StateChart



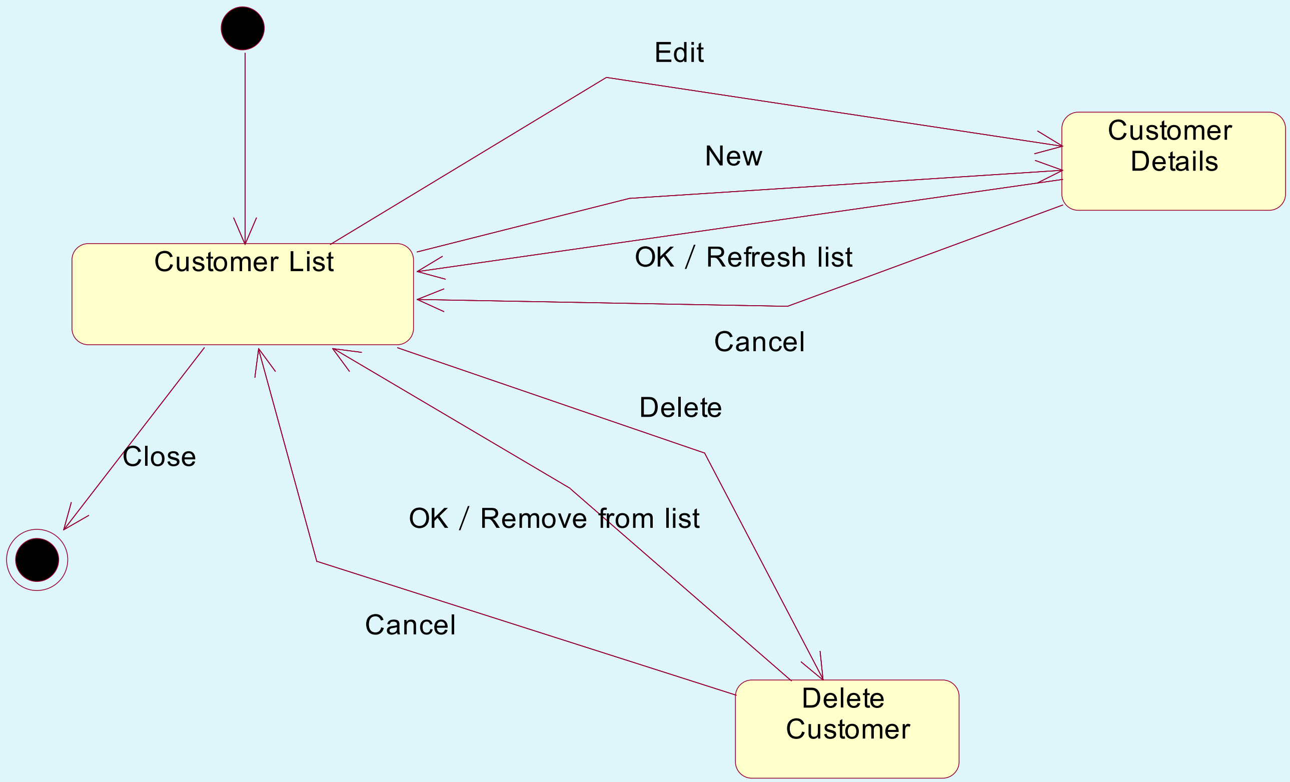
All rights reserved to D.Maymone

State with Activities



All rights reserved to D.Maymone

Interaction Between Boundary Classes



All rights reserved to D.Maymone

תרגיל ב- State Diagram

- בנה State Diagram עבור שעון דיגיטלי.
- לשעון הכפתורים הבאים
  - כפתור לשינוי סוג התצוגה (רגיל, שינוי שעות, שינוי דקות, שינוי שניות)
  - כפתור להגדלת המספר
  - כפתור להקטנת המספר
- בנה גם את ה- Class של השעון

כל הזכויות שמורות לד. מימון ייעוץ והדרכה במח[illegible] בע"מ

בדיקת המודל

- שלמות
- מצבים לא מופעלים – Unreachable States
- מצבים שלא ניתן לצאת מהם – Dead States

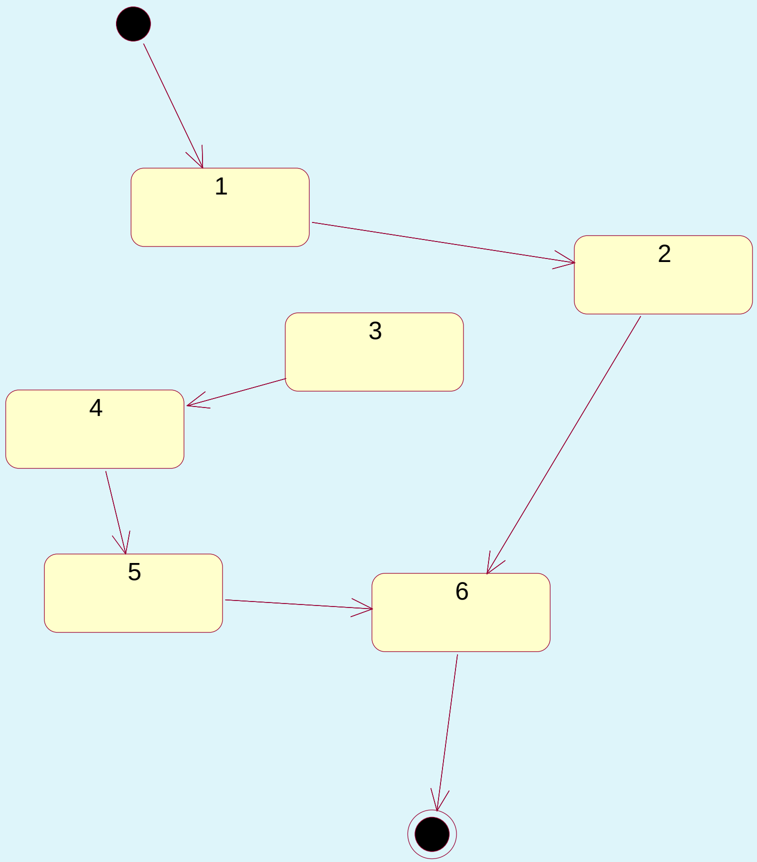
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במח[illegible] בע"מ

שלמות

- האם כל צירוף של מצב/ארוע מטופל?
- עבור כל transition בדוק אם כל הקלטים מטופלים במצב הבא. אם נמצא קלט לא מטופל כנראה שיש ליצור מצב נוסף
- תעד Transitions כגון מצבי שגיאה

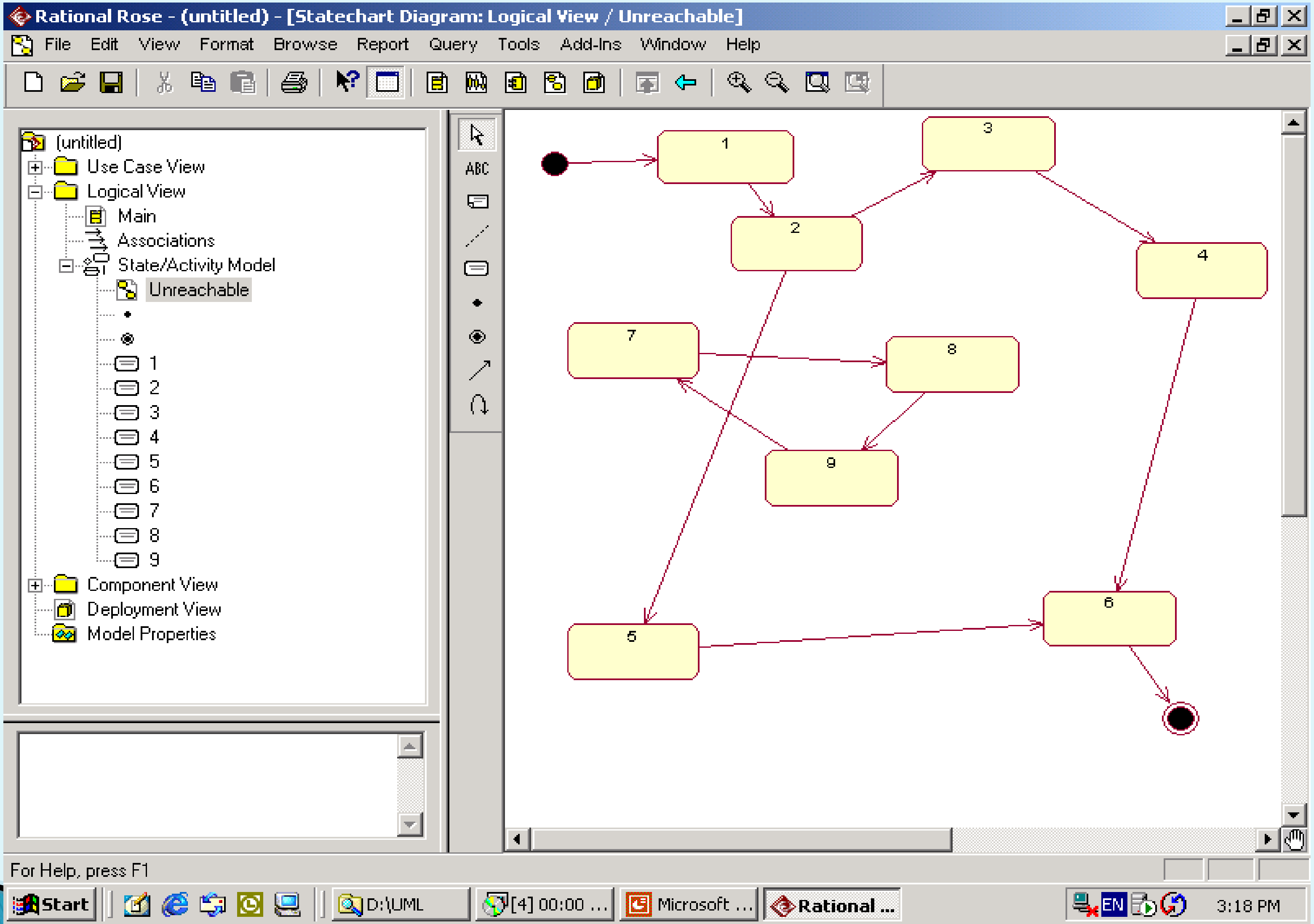
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במחשוב בע"מ

Unreachable States



All rights reserved to D.Maymone

Dead States



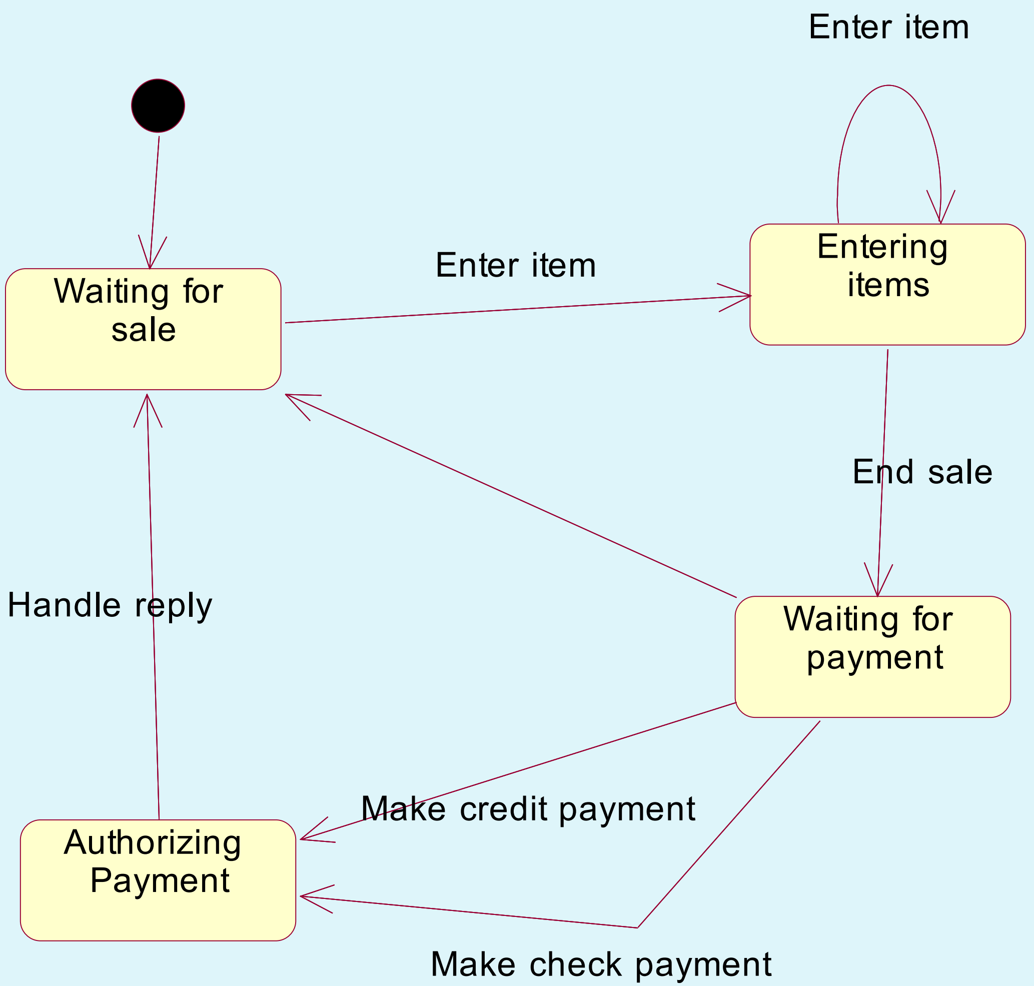
כל הזכויות שמורות לד. מימון ייעוץ והדרכה במ ם בע"מ

Examples

- During the "Process Document" use case in a word processor, it is not legal to perform the File-Save operation until the File-New or File-Open event has happened
- Windows:
  - Edit-Paste action is only valid if there is something in the "clipboard" to paste
- Compilers
- Editors
- Controllers
- Devices

All rights reserved to D.Maymone

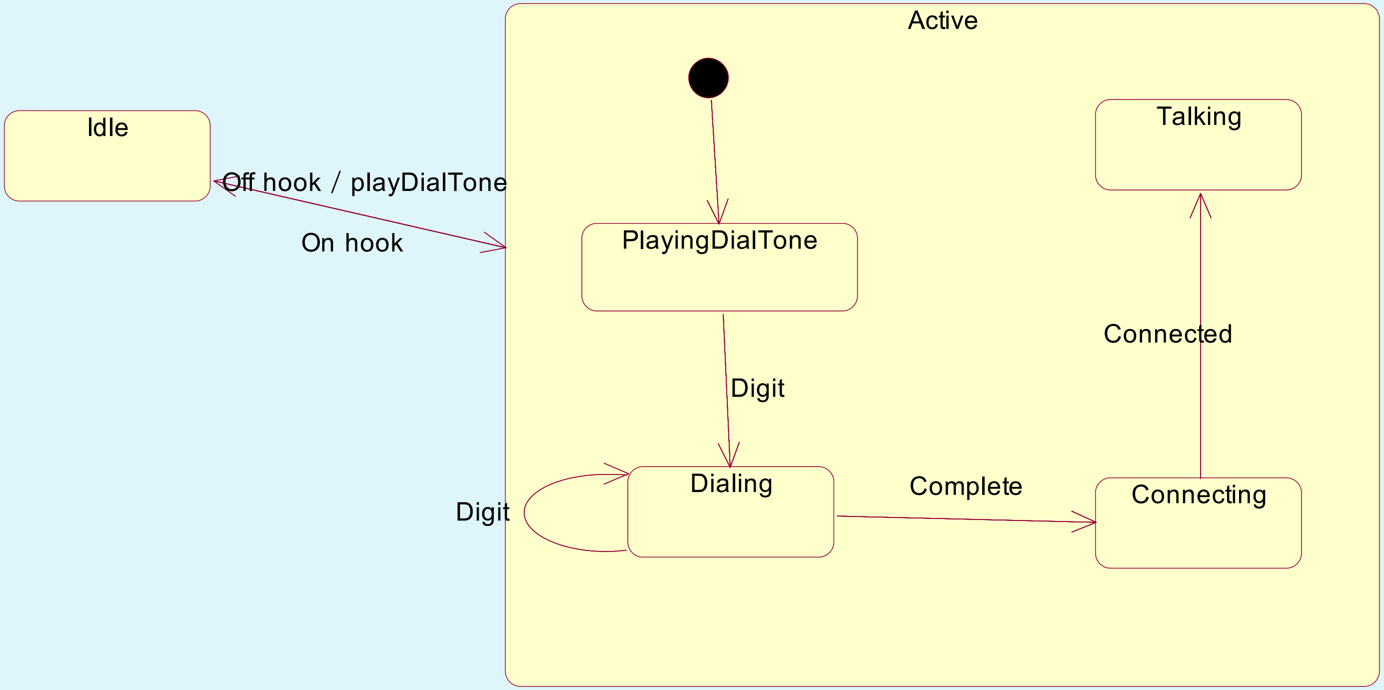
Buy Items



All rights reserved to D.Maymone

Nested States

- ניתן לבנות דיאגרמת מצבים בתוך כל מצב



All rights reserved to D.Maymone

טבלאות מצבים

- מבוא
- דוגמת ה- MOVE
- דיאגרמת מצבים של ה- MOVE
- טבלת מצבים ללא פונקציות
- טבלת מצבים עם פונקציות
- האלגוריתם של טבלת המצבים
- תרגילים

All rights reserved to D.Maymone

דוגמת פקודת ה- MOVE של COBOL

Build an algorith to validate the MOVE COBOL statement.
The valid syntax is as follows:

MOVE { constant/variable } TO VAR_1 VAR_2 …. VAR_N

Invalid examples:

MOVE A TO 3
MOVE TO A B
TO A MOVE B
A TO MOVE X

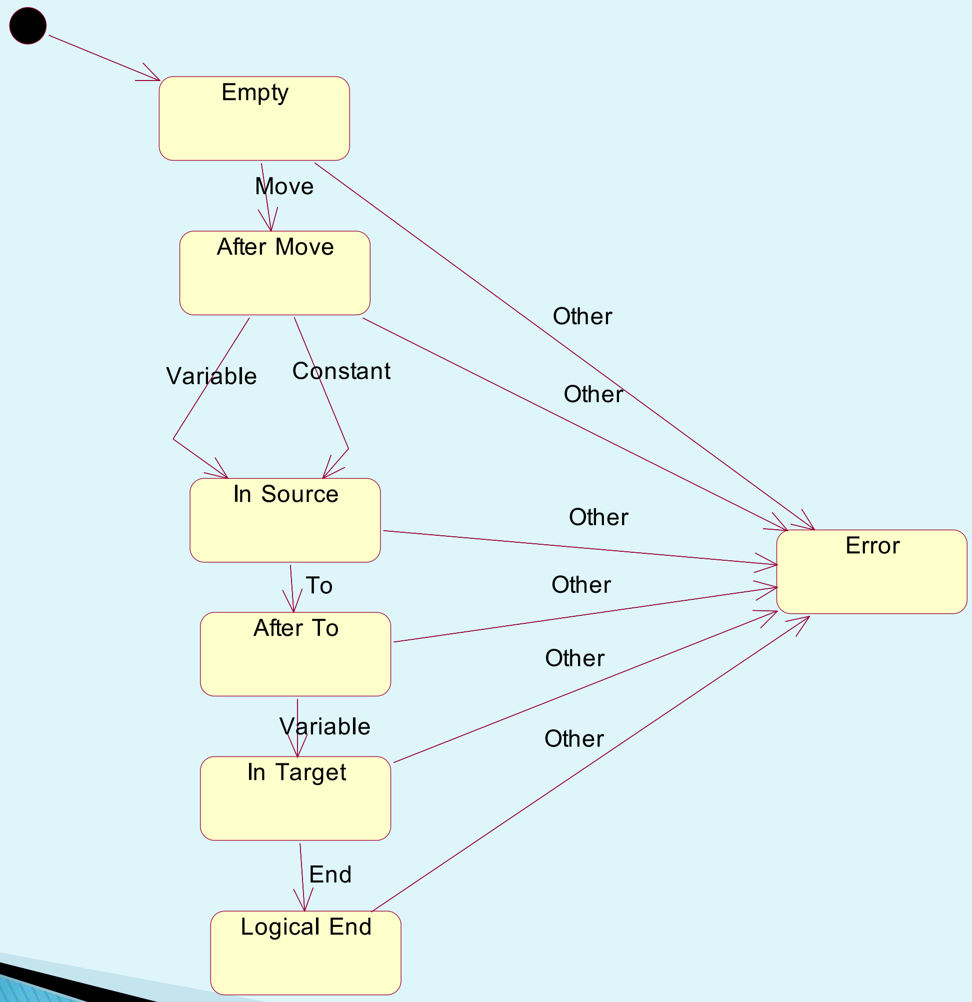
Assume that a table containing the different components of the statement already exists (T(20))

- T -

MOVE
A
TO
S
T
U
…..
…..
…..

All rights reserved to D.Maymone

Move Statement – Statechart Diagram



All rights reserved to D.Maymone

טבלת מצבים ללא פונקציות

- TS -

Input / State	MOVE	Constant	Variable	TO	End
1 - Empty	2	-1	-1	-1	-1
2 – After MOVE	-2	3	3	-3	-4
3 – In Source	-2	-5	-5	4	-4
4 – After TO	-2	-6	5	-7	-4
5 – In Target	-2	-6	5	-7	6
6 – Logical End	-8	-8	-8	-8	

1. Statement does not start with MOVE clause
2. Duplicate MOVE clause
3. Missing source clause
4. Incomplete statement
5. Duplicate source clause
6. A constant is not allowed as target
7. Duplicate TO clause
8. Components appear after end of input

All rights reserved to D.Maymone

טבלת מצבים עם פונקציות

- נתונות רשומות עם הסוגים הבאים: R1, R2, R3
- אם R1 מופיע אחרי R2 בצע P1
- אם R3 מופיע אחרי מספר זוגי של R2 בצע P3
- אם R3 מופיע אחרי מספר אי-זוגי של R2 בצע P4
- אם R1 מופיע בצורה רגילה בצע P5
- אם R2 מופיע בצורה רגילה בצע P6
- אם R3 מופיע בצורה רגילה בצע P7

All rights reserved to D.Maymone

דוגמה

- R1 → P5
- R1 → P5
- R2 → P6
- R3 → P4
- R1 → P2
- R2 → P6
- R2 → P6
- R3 → P3
- R2 → P6
- R1 → P1
- R3 → P7

All rights reserved to D.Maymone

States and Functions in the Table

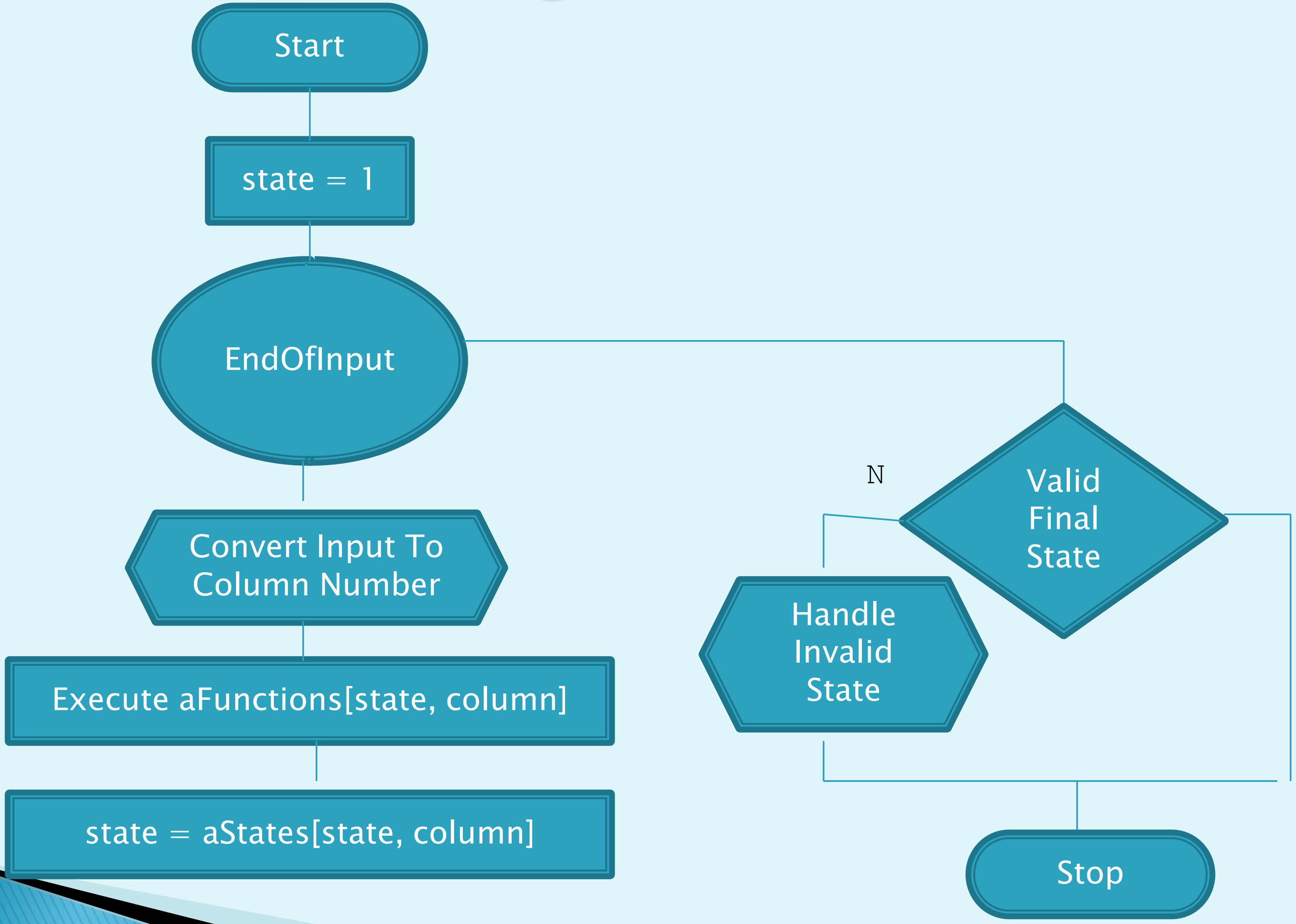
Input State	R1	R2	R3
1 - Empty	P5/2	P6/4	P7/5
2 - R1	P5/2	P6/4	P7/5
3 - R2 odd	P1/2	P6/4	P3/5
4 - R2 even	P1/2	P6/3	P4/5
5 - R3	P2/2	P6/4	P7/5

Function

State

All rights reserved to D.Maymone

State Table Algorithm



All rights reserved to D.Maymone

Constant Number Exercise

- Given a variable 20 chars long. Check if its content contains a valid presentation of a numeric constant.
- Valid Structure: Sddd-----d.dddETddd
- S – is optional
- If E exist the T is mandatory
- Check the following conditions
  - There is no spaces between S and the first digit
  - The integer part of the number contains 1 to 10 digits
  - The fraction part of the number contains up to 5 digits
  - The exponent contains 1 to 3 digits
- Valid examples: 123, -24, -127.39, +12345, -123.45E+3, 123.3e-12
- Invalid examples: + 123, 123.45.67, 123-, +-12.34, 15o.7, 123$

All rights reserved to D.Maymone

Constant Number

Input State	Space				
Empty					

All rights reserved to D.Maymone

SQL Statement

- Create a statechart diagram for the SQL SELECT statement

```
* FROM  t1 [,t2] [,t3] ...
   Or
SELECT col1/lit1 [, col2/lit2] [,
   col3/lit3] FROM  t1 [,t2] [,t3] ...
[WHERE cond1 [AND/OR cond2 ] [AND/OR
   cond3] ...]
[ORDER BY col1/const1 [, col2/const2]
   [, col3/const3] ...
```

All rights reserved to D.Maymone

Input

- **Valid**
- Select
- *
- From
- Table
- Table
- Table
- Where
- Condition
- And
- Condition

- **Invalid**
- Select
- Column
- Column
- Column
- From
- Table
- Table
- Table
- Where
- Condition
- And
- Table
- Condition

- **Invalid**
- Select
- *
- Column
- From
- Table
- Column
- Table
- Where
- Condition
- And
- Condition

All rights reserved to D.Maymone

בעית התשדורות

- נתונה תשדורת של אפסים ואחדים
- בתחילת התשדורת מופיע A ובסיומה Z
- בנה טבלת מצבים לתוכנית אשר תקבל בהדרגה תווים מתקן חיצוני ותציג הודעה כל פעם שהופיעה התשדורת 101.
- לדוגמה
- A11010001000101010Z

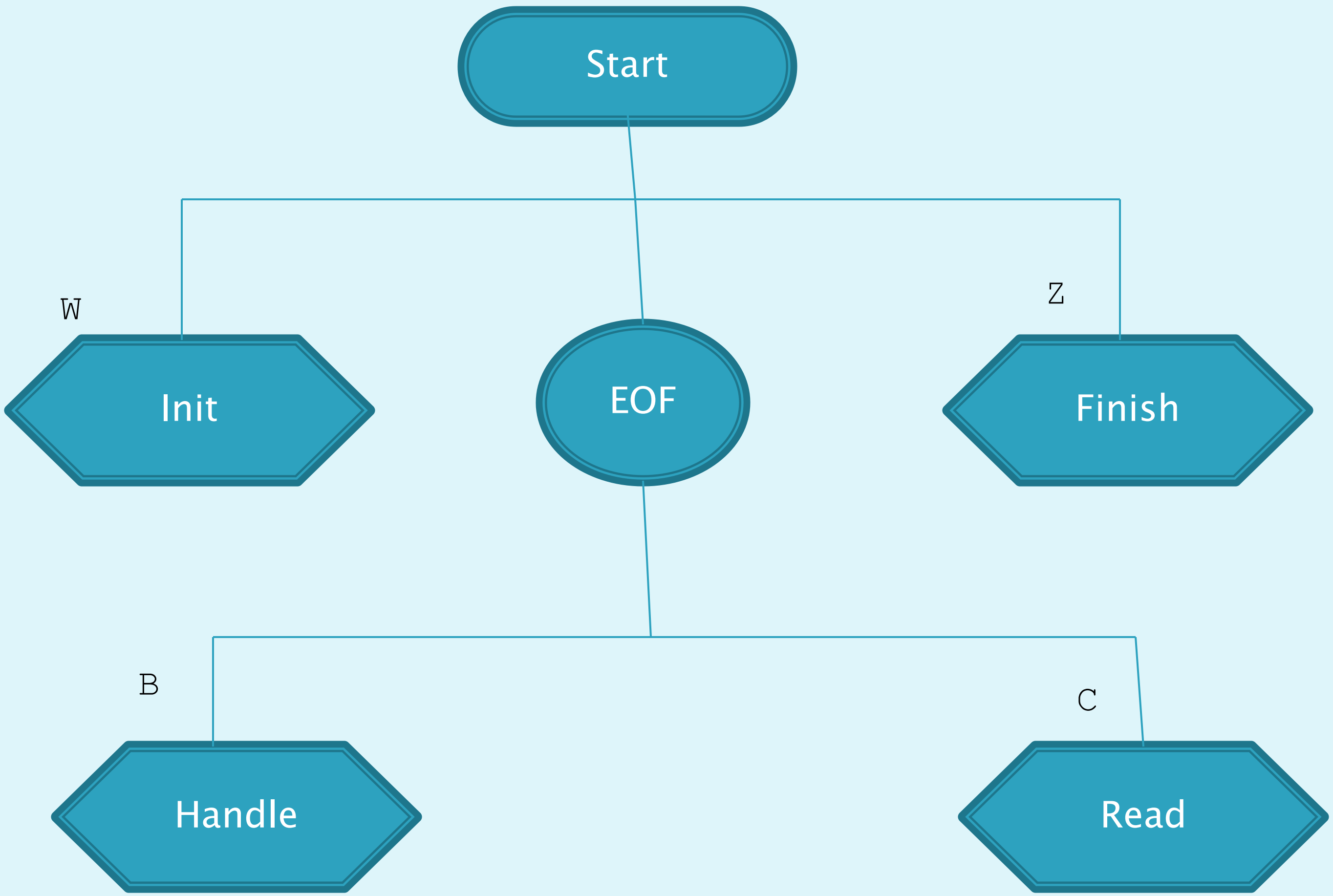
All rights reserved to D.Maymone

מודלים לתכנות

- מודל סדרתי
- מודל שבירה
- מודל הקבלה
- מודל מיזוג - Syncsort
- מודל מיזוג עם שבירה - Syncsort

All rights reserved to D.Maymone

מודל סדרתי



All rights reserved to D.Maymone

Cobol Example

```
03 SW01-I01-STATUS   PIC S9(4) BINARY  VALUE 0.
      88 SW01-I01-END-OF-FILE          VALUE 9.
…………………………………………..
*----------------------------------
  A-MAIN                    SECTION.
*----------------------------------
  A00.
      PERFORM W-INIT
      PERFORM UNTIL SW01-I01-END-OF-FILE
             PERFORM B-HANDLE
             PERFORM C-READ
      END-PERFORM
      PERFORM Z-FINISH
      GOBACK.
  A-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*----------------------------------------
  C-READ                          SECTION.
*----------------------------------------
  C00.
      READ I01-EMPLOYEES
            AT END
                  SET SW01-I01-END-OF-FILE TO TRUE
      END-READ.
  C-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

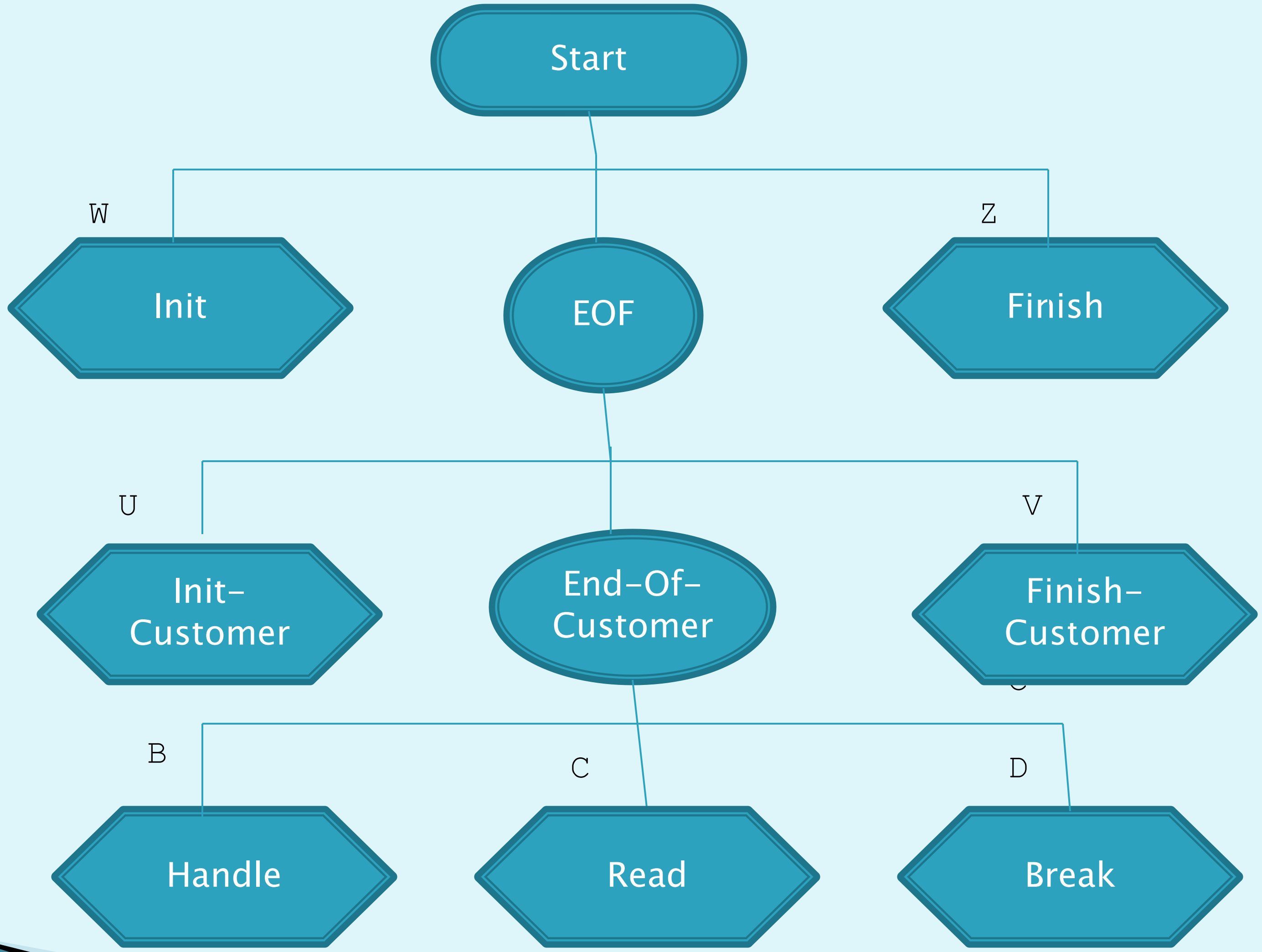
```
*---------------------------------------
  W-INIT                          SECTION.
*---------------------------------------
  W00.
      OPEN INPUT I01-EMPLOYEES
      IF NOT WS02-NORMAL-STATUS
             SET WS04-FILE-NOT-FOUND TO TRUE
             PERFORM X-ERROR
      END-IF
```

- Read first record

```
      PERFORM C-READ
      IF SW01-I01-END-OF-FILE
             SET WS04-EMPTY-FILE TO TRUE
             PERFORM X-ERROR
      END-IF.
  W-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

מודל שבירה עם מפתח אחד



All rights reserved to D.Maymone

Cobol Example

```
03 SW01-I01-STATUS   PIC S9(4) BINARY     VALUE 0.
        88 SW01-I01-NORMA;-STATUS         VALUE 0.
        88 SW01-I01-END-OF-CUSTOMER       VALUE 6 THRU 9.
        88 SW01-I01-END-OF-FILE           VALUE 9.
……………………………………………..
*--------------------------------
  A-MAIN                SECTION.
*--------------------------------
  A00.
        PERFORM W-INIT
        PERFORM UNTIL SW01-I01-END-OF-FILE
                PERFORM U-INIT-CUSTOMER
                PERFORM UNTIL SW01-I01-END-OF-CUSTOMER
                        PERFORM B-HANDLE
                        PERFORM C-READ
                        PERFORM D-VREAK
                END-PERFORM
                PERFORM V-FINISH-CUSTOMER
        END-PERFORM
        PERFORM Z-FINISH
        GOBACK.
  A-EXIT.
        EXIT.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*---------------------------------------
  U-INIT-CUSTOMER        SECTION.
*---------------------------------------
  U00.
```

- Save Customer id in previous

```
      SET SW01-I01-NORMAL-STATUS TO TRUE
      MOVE I01-CUSTOMER-ID TO
                    WS02-PREVIOUS-CUSTOMER-ID
  U-EXIT.
      EXIT.
```

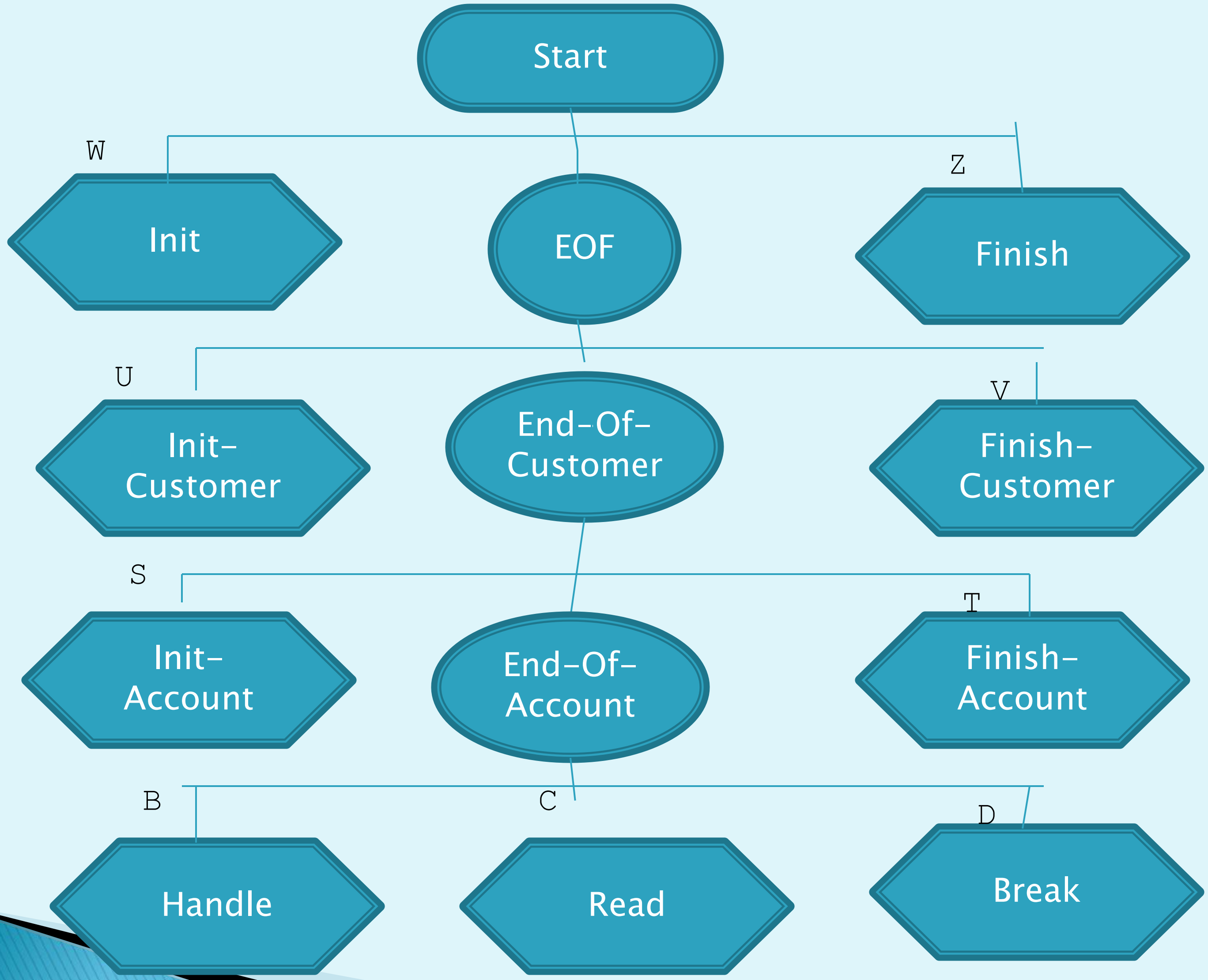
All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*--------------------------------
  D-BREAK                    SECTION.
*--------------------------------
  D00.
      IF I01-CUSTOMER-ID NOT =
                   WS02-PREVIOUS-CUSTOMER-ID
            SET SW01-I01-END-OF-CUSTOMER TO TRUE
      END-IF
  D-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

מודל שבירה עם שני מפתחות או יותר



All rights reserved to D.Maymone

Cobol Example

```
03 SW01-I01-STATUS   PIC S9(4) BINARY     VALUE 0.
       88 SW01-I01-NORMAL-STATUS          VALUE 0.
       88 SW01-I01-END-OF-ACCOUNT         VALUE 3 THRU 9.
       88 SW01-I01-END-OF-CUSTOMER        VALUE 6 THRU 9.
       88 SW01-I01-END-OF-FILE            VALUE 9.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

```
A00.
      PERFORM W-INIT
      PERFORM UNTIL SW01-I01-END-OF-FILE
          PERFORM U-INIT-CUSTOMER
          PERFORM UNTIL SW01-I01-END-OF-CUSTOMER
              PERFORM S-INIT-ACCOUNT
              PERFORM UNTIL SW01-I01-END-OF-ACCOUNT
                       PERFORM B-HANDLE
                       PERFORM C-READ
                       PERFORM D-BREAK
               END-PERFORM
               PERFORM T-FINISH-ACCOUNT
          END-PERFORM
          PERFORM V-FINISH-CUSTOMER
      END-PERFORM
      PERFORM Z-FINISH
      GOBACK.
A-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*-------------------------------------
  U-INIT-CUSTOMER       SECTION.
*-------------------------------------
  U00.
      SET SW01-I01-NORMAL-STATUS TO TRUE
*     Save Customer id in previous
      MOVE I01-CUSTOMER-ID TO
                    WS02-PREVIOUS-CUSTOMER-ID.
  U-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*--------------------------------------
  S-INIT-ACCOUNT         SECTION.
*--------------------------------------
  S00.
      SET SW01-I01-NORMAL-STATUS TO TRUE
*     Save Account id in previous
      MOVE I01-ACCOUNT-ID TO
                    WS02-PREVIOUS-ACCOUNT-ID.
  S-EXIT.
      EXIT.
```

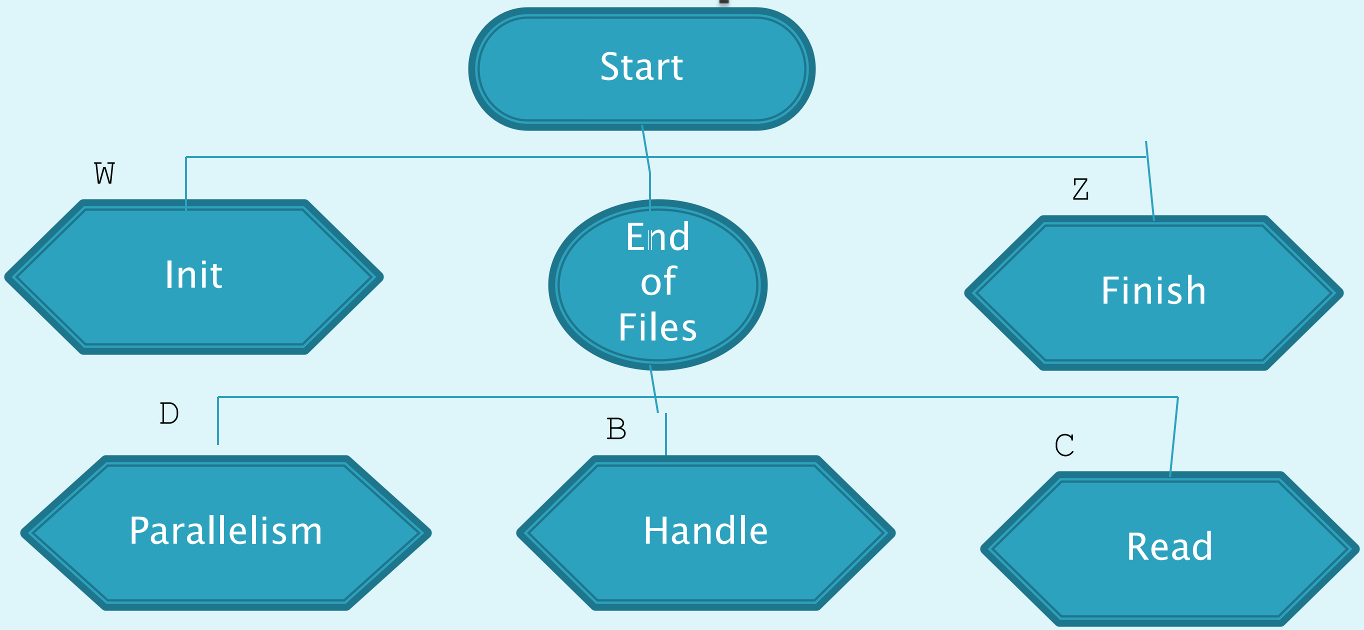
All rights reserved to D.Maymone

Cobol Example (cont'd)

```
*---------------------------------
  D-BREAK                        SECTION.
*---------------------------------
  D00.
      EVALUATE TRUE
             WHEN I01-CUSTOMER-ID NOT =
                    WS02-PREVIOUS-CUSTOMER-ID
                    SET SW01-I01-END-OF-CUSTOMER TO TRUE
             WHEN I01-ACCOUNT-ID NOT =
                    WS02-PREVIOUS-ACCOUNT-ID
                    SET SW01-I01-END-OF-ACCOUNT TO TRUE
             WHEN OTHER
                    EXIT SECTION
      END-EVALUATE.
  D-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

מודל הקבלה



All rights reserved to D.Maymone

Cobol Parallelism

```
*-----------------------------------------------
 A-MAIN                               SECTION.
*-----------------------------------------------
 A00.
     PERFORM W-INIT
     PERFORM UNTIL SW01-I01-END-OF-FILE
         PERFORM D-PARALLELISM
         PERFORM B-HANDLE
         PERFORM C-READ-MAIN
     END-PERFORM
     PERFORM Z-FINISH
     GOBACK.
 A-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Parallelism(cont'd)

```
*-----------------------------------------
 B-HANDLE                        SECTION.
*------------------------------------------
 B00.
     IF I01-KEYS NOT = I02-KEYS
         GO TO D-EXIT
     END-IF.
*     Add here your functionality **
 B-EXIT.
     EXIT.
```

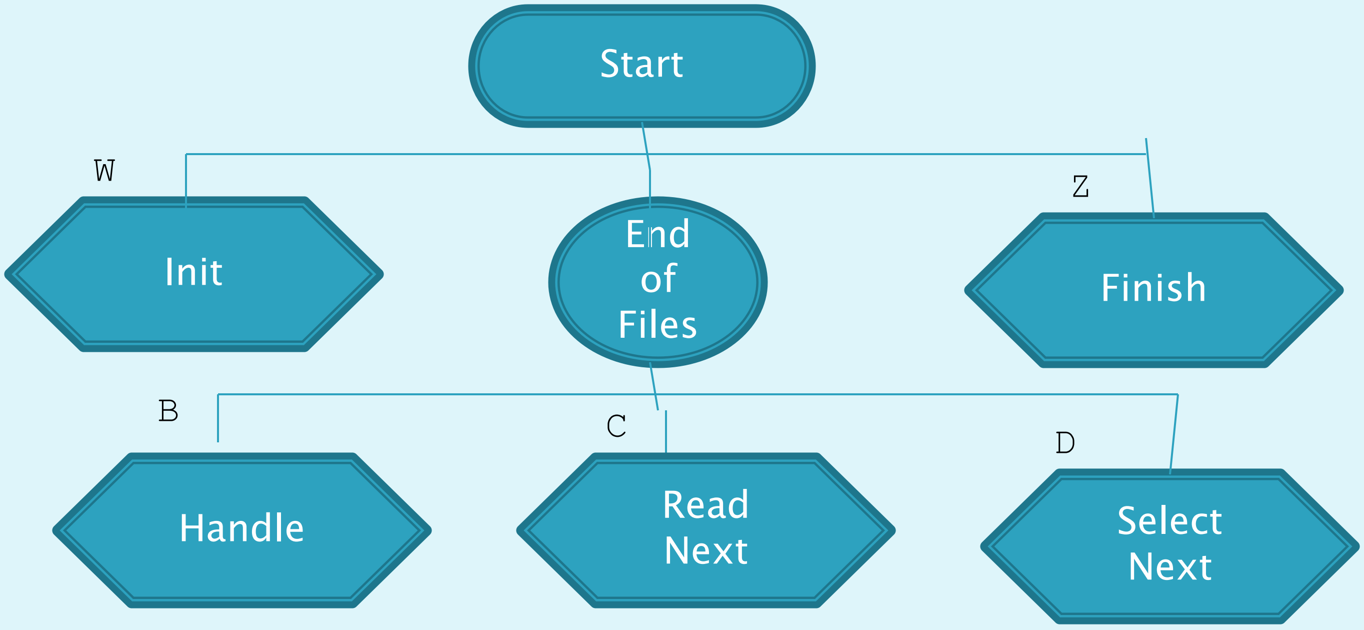
All rights reserved to D.Maymone

Cobol Parallelism(cont'd)

```
*----------------------------------------
 D-PARALLELISM                    SECTION.
*----------------------------------------
 D00.
     PERFORM UNTIL I02-KEYS >= I01-KEYS
         READ I02-SECONDAY
              AT END
                  MOVE HIGH-VALUE TO I02-KEYS
         END-READ
     END-PERFORM.
 D-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

מודל מיזוג



All rights reserved to D.Maymone

Cobol Merge

```
*-----------------------------------------------------------
IXXX - INDEXES
*----------------------------------------------------------
  01  IX00.
      03 IX01-CURRENT-FILE  PIC S9(4) BINARY.
*-------------------------------------------------------
* TBXX - TABLES
*-------------------------------------------------------
  01 TB01-FILES.
      03 TB01-FILE      OCCURS 0 TO 3
                       DEPENDING CO01-NUMBER-OF-FILES
                       INDEXED BY TB01-I.
            05 TB01-ID PIC S9(4).
```

All rights reserved to D.Maymone

Cobol Merge (cont'd)

```
*----------------------------------------------------------
  A-MAIN                                 SECTION.
*----------------------------------------------------------
  A-00.
      PERFORM W-INIT
      PERFORM UNTIL SW01-END-OF-FILES
            PERFORM B-HANDLE
            PERFORM C-READ-NEXT
            PERFORM D-SELECT-NEXT
      END-PERFORM
      PERFORM Z-FINISH.
      GOBACK.
  A-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Merge (cont'd)

```
*-----------------------------------------------------------------
  C-READ-NEXT                          SECTION.
*-----------------------------------------------------------------
  C00.
      EVALUATE TRUE
               WHEN IX01-CURRENT-FILE = CO02-I01-FILE-NUMBER
                       PERFORM CA-READ-I01
               WHEN IX01-CURRENT-FILE = CO02-I02-FILE-NUMBER
                       PERFORM CB-READ-I02
               WHEN IX01-CURRENT-FILE = CO02-I03-FILE-NUMBER
                       PERFORM CC-READ-I03
               WHEN OTHER
                       SET WS02-INVALID-FILE-NUMBER TO TRUE
                       PERFORM X-ERROR
      END-EVALUATE.
  C-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Merge (cont'd)

```
*-----------------------------------------------------
  D-SELECT-NEXT                                 SECTION.
*-----------------------------------------------------
  D00.
      MOVE FUNCTION ORD-MIN(TB01-ID(ALL)) TO
                  IX01-CURRENT-FILE
      IF FUNCTION MIN(TB01-ID(ALL)) = CO03-HIGH-VALUE
            SET SW01-END-OF-FILES TO TRUE
      END-IF.
  D-EXIT.
      EXIT.
```

All rights reserved to D.Maymone

Cobol Merge (cont'd)

```
*---------------------------------------------------------
  W-INIT                                SECTION.
*---------------------------------------------------------
*    Open files
*     .........
*     Read one record from each file
     PERFORM C-READ-NEXT VARYING IX01-CURRENT-FILE
           FROM 1
           BY 1
           UNTIL  IX01-CURRENT-FILE >
                                 CO01-NUMBER-OF-FILES
     PERFORM D-SELECT-NEXT.
  W-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

מודל מיזוג עם שבירה

- נתונים 3 קבצים ממוינים לפי מספר לקוח
  - קובץ לקוחות
  - קובץ מכתבים ללקוחות
  - קובץ תשובות
- יש להדפיס את הדוחות הבאים:
  - לקוחות שלא נשלחו אליהם מכתבים
  - לקוחות שקיבלו מכתב ולא ענו
  - לקוחות שקיבלו מכתב וענו
  - לקוחות שלא קיבלו מכתב ענו

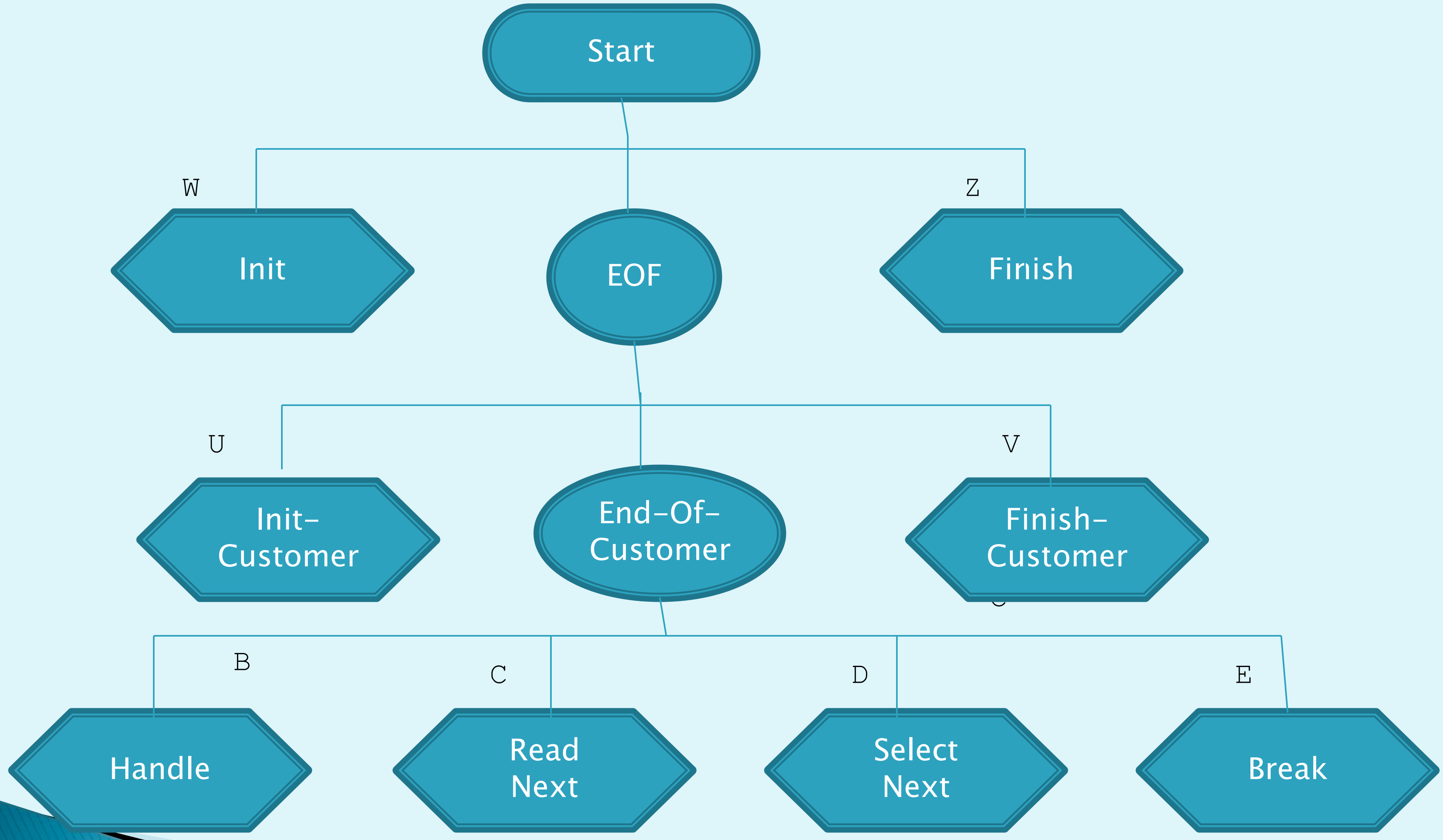
All rights reserved to D.Maymone

דוגמת לקוחות ומכתבים

תשובות	מכתבים	לקוחות
42	17	13
47	42	17
47	42	25
63	42	42
65	59	58
65	63	63
	65	67
	87	81
		87
		91
		93

All rights reserved to D.Maymone

מודל מיזוג עם שבירה



All rights reserved to D.Maymone

Cobol Merge And Break

```
01 TB01-FILES.
           03 TB01-FILE                OCCURS 0 TO 5
                                       DEPENDING WS03-NUMBER-OF-ENTRIES
                                       INDEXED BY TB01-I.
              05 TB01-ID               PIC S9(4).
      *--------------------------------------+----------|
      * Seq. | Answers | Letters | Customers | Reports  |
      *--------------------------------------+----------|
      * 1.   |    0    |    0    |     0     |          |
      * 2.   |    0    |    0    |     1     | 1        |
      * 3.   |    0    |    1    |     0     |          |
      * 4.   |    0    |    1    |     1     | 2        |
      * 5.   |    1    |    0    |     0     |          |
      * 6.   |    1    |    0    |     1     | 1+4      |
      * 7.   |    1    |    1    |     0     |          |
      * 8.   |    1    |    1    |     1     | 3        |
      *--------------------------------------+----------|
       01  TB02-REPORTS     VALUE  '   1     2     1+4   3   '.
03 TB02-REPORT              OCCURS 0 TO 8
                            DEPENDING CO04-NO-OF-DECISIONS
                            INDEXED BY TB02-I
                            PIC X(3).
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 A-MAIN                              SECTION.
*-----------------------------------------------------------------
 A-00.
     PERFORM W-INIT
     PERFORM UNTIL SW01-END-OF-FILES
         PERFORM U-INIT-CUSTOMER
         PERFORM UNTIL SW01-END-OF-CUSTOMER
             PERFORM B-HANDLE
             PERFORM C-READ-NEXT
             PERFORM D-SELECT-NEXT
             PERFORM E-BREAK
         END-PERFORM
         PERFORM V-FINISH-CUSTOMER
     END-PERFORM
     PERFORM Z-FINISH.
     GOBACK.
 A-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 B-HANDLE                                SECTION.
*-----------------------------------------------------------------
 B00.
     EVALUATE TRUE
         WHEN IX01-CURRENT-FILE = CO02-I01-FILE-NUMBER
             SET SW02-CUSTOMER-FOUND TO TRUE
         WHEN IX01-CURRENT-FILE = CO02-I02-FILE-NUMBER
             SET SW02-LETTER-FOUND TO TRUE
         WHEN IX01-CURRENT-FILE = CO02-I03-FILE-NUMBER
             SET SW02-ANSWER-FOUND TO TRUE
         WHEN OTHER
             NEXT SENTENCE
     END-EVALUATE.
 B-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 C-READ-NEXT                          SECTION.
*-----------------------------------------------------------------
 C00.
     EVALUATE TRUE
         WHEN IX01-CURRENT-FILE = CO02-I01-FILE-NUMBER
             PERFORM CA-READ-I01
         WHEN IX01-CURRENT-FILE = CO02-I02-FILE-NUMBER
             PERFORM CB-READ-I02
         WHEN IX01-CURRENT-FILE = CO02-I03-FILE-NUMBER
             PERFORM CC-READ-I03
         WHEN OTHER
             SET WS02-INVALID-FILE-NUMBER TO TRUE
             PERFORM X-ERROR
     END-EVALUATE.
 C-EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 D-SELECT-NEXT                     SECTION.
*-----------------------------------------------------------------
 D00.
*    fIND THE MINIMAL id
     MOVE TB01-ID(1) TO TB01-ID(WS03-NUMBER-OF-ENTRIES)
     MOVE 1 TO IX01-CURRENT-FILE
     PERFORM VARYING TB01-I
             FROM 2
             BY 1
             UNTIL TB01-I > CO01-NUMBER-OF-FILES
         IF TB01-ID(TB01-I) < TB01-ID(WS03-NUMBER-OF-ENTRIES)
             MOVE TB01-ID(TB01-I) TO
                                 TB01-ID(WS03-NUMBER-OF-ENTRIES)
             SET IX01-CURRENT-FILE TO TB01-I
         END-IF
     END-PERFORM.
 D-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 E-BREAK                                SECTION.
*-----------------------------------------------------------------
 E00.
     EVALUATE TRUE
          WHEN TB01-ID(WS03-NUMBER-OF-ENTRIES) = CO03-HIGH-VALUE
               SET SW01-END-OF-FILES TO TRUE
          WHEN TB01-ID(WS03-NUMBER-OF-ENTRIES) NOT =
                    TB01-ID(WS03-NUMBER-OF-ENTRIES - 1)
               SET SW01-END-OF-CUSTOMER TO TRUE
          WHEN OTHER
               SET WS02-I01-FILE-NOT-FOUND TO TRUE
               PERFORM X-ERROR
     END-EVALUATE.
 E-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*-----------------------------------------------------------------
 U-INIT-CUSTOMER                  SECTION.
*-----------------------------------------------------------------
 U00.
*    SAVE THE CURRENT CUSTOMER IN PREVIOUS
     MOVE TB01-ID(WS03-NUMBER-OF-ENTRIES) TO
             TB01-ID(WS03-NUMBER-OF-ENTRIES - 1)
     SET SW01-NORMAL-STATUS      TO TRUE
     SET SW02-CUSTOMER-INIT      TO TRUE
     SET SW02-LETTER-INIT        TO TRUE
     SET SW02-ANSWER-INIT        TO TRUE.
 U-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*------------------------------------------------------------------
 V-FINISH-CUSTOMER                SECTION.
*------------------------------------------------------------------
 V00.
*    COMPUTE THE DECISION TABLE ENTRY
     COMPUTE IX02-ENTRY = (SW02-CUSTOMER-STATUS *
                                 CO05-CUSTOMER-MULTIPLIER) +
                          (SW02-LETTER-STATUS *
                                 CO05-LETTER-MULTIPLIER) +
                          (SW02-ANSWER-STATUS *
                                 CO05-ANSWER-MULTIPLIER) +
                  1
     IF TB02-REPORT(IX02-ENTRY) = SPACES
         EXIT SECTION
     END-IF
     MOVE TB01-ID(WS03-NUMBER-OF-ENTRIES - 1) TO O01-ID
     MOVE CO06-REPORTS-TITLE TO O01-DETAILS.
     MOVE TB02-REPORT(IX02-ENTRY) TO O01-DETAILS (11:3)
     WRITE O01-R-FD.
 V-EXIT.
     EXIT.
```

All rights reserved to D.Maymone

Cobol Merge And Break (cont'd)

```
*----------------------------------------------------------------
 W-INIT                                 SECTION.
*----------------------------------------------------------------
  W00.
 *     Open Files
        COMPUTE WS03-NUMBER-OF-ENTRIES = CO01-NUMBER-OF-FILES + 2
*      Read one record from each file
       PERFORM C-READ-NEXT VARYING IX01-CURRENT-FILE
                           FROM 1
                           BY 1
                           UNTIL  IX01-CURRENT-FILE >
                                        CO01-NUMBER-OF-FILES
       PERFORM D-SELECT-NEXT.
 W-EXIT.
       EXIT.
```

All rights reserved to D.Maymone

טיפול בטבלאות/מערכים

- שימוש באינדקסים
- אתחול טבלאות
- שיטות חיפוש
- שיטות מיון מערכים

All rights reserved to D.Maymone

שימוש באינדקסים

- הבעיה: כתובת הכניסה במערך מחושבת בזמן ריצה ולא ע"י הקומפיילר
- הצורה היעלה ביותר - קבועים
- אינדקסים חופשיים יוגדרו : PIC S9(4) BINARY
- אינדקסים קשורים לטבלה יוגדרו ע"י INDEXED BY
- כאשר יש יותר מאשר פניה אחת למשתנה עם אינדקס, יש להעבירו לשטח עזר
- השמה למשתנה בדיד
- השמה לכניסה הראשונה
- השמה לכניסה האחרונה + 1
- הערה: אם האינדקס הוא קבוע הכתובת מחושבת בזמן קומפילציה

All rights reserved to D.Maymone

מה הבעיה בקטע הבא?

```
IF T(I) > B
  ADD T(I) TO C
  COMPUTE X = T(I) + W
  MOVE T(I) TO B
  SUBTRACT T(I) FROM Z
END-IF
```

All rights reserved to D.Maymone

נתונה הטבלה

	ID מס' עובד	GR דרגה	C ילדים	BS מש. יסוד	G מין	P מקצוע	E ותק	A גיל	T סוג עובד
1									
2									
3									
200									

All rights reserved to D.Maymone

מה הבעיה בקטע הבא?

```
IF TB01-GRADE(I) > 8 AND
           TB01-CHILDREN(I) > 3 AND < 7 AND
           (TB01-EXPERIENCE(I) > 5 OR
                       TB01-AGE(I) > 37) AND
           TB01-TYPE(I) = 3 AND
           TB01-BASIC-SALARY(I) > 6000 AND < 15000
  …
  …
  …
END-IF
```

All rights reserved to D.Maymone

הגדרת פרוט הכניסה לפני הטבלה

```
01  TB01-EMPLOYEES.
      03 TB01-ROW.
             05  TB01-ID
             05     TB01-GRADE
             05     TB01-CHILDREN
             05     TB01-BASIC-SALARY
             05     TB01-GENDER
             05     TB01-PROFESSION
             05     TB01-EXPERIENCE
             05     TB01-AGE
             05     TB01-TYPE
      03 TB01-EMPLOYEE OCCURS 0 TO 200
                     DEPENDING ON WS01-NUMBER-OF-EMPLOYEES
                     INDEXED BY TB01-I.
             05 TB01-EMPLOYEE-ID
             05 FILLER
```

All rights reserved to D.Maymone

אתחולים

- יש להשתמש באתחולים מיוחדים כאשר מתבצע אתחול מספר פעמים.
- איפוס בדיד של רכיבים
- איפוס שורות
- איפוס טבלאות בינריות ע"י LOW-VALUE
- איפוס לוגי ע"י pointer
- איפוס ע"י שטח קבוע
- איפוס ע"י מריחה
- אם כל שורה מכילה הרבה עמודות, עדיף לאפס את השורה הראשונה ולהעתיק אותה לשאר השורות
- שמור את הטבלה המאותחלת בשטח אחר והשתמש בו לאתחולים הבאים של אותה טבלה.

All rights reserved to D.Maymone

דוגמה לא נכונה

```
01  TB01-ITEMS.
      03 TB01-ITEM        OCCURS 100.
            05    TB01-ID                  PIC 9(4) COMP.
            05    TB01-QUANTITY            PIC 9(5).
            05    TB01-PRICE               PIC S9(9)V99.
            05    TB01-VAT                 PIC S9(8)V99.

      MOVE ZERO TO TB01-ITEMS
```

All rights reserved to D.Maymone

איפוס בדיד

```
01  TB01-BRANCHES.
      03 TB01-BRANCH                             OCCURS 100
                                                 INDEXED BY TB01-I.
            05     TB01-A                        PIC 9(5).
            05     TB01-B                        PIC 9(9) COMP-3.
            05     TB01-C                        PIC X(20).
            05     TB01-D                        PIC S9(8) BINARY.
            05     TB01-E                        PIC X(5).
      PERFORM VARYING TB01-I FROM 1 BY 1
                   UNTIL TB01-I > 100
            MOVE ZERO TO TB01-A(TB01-I)
                         TB01-B(TB01-I)
                         TB01-D(TB01-I)
            MOVE SPACES TO TB01-C(TB01-I)
                               TB01-E(TB01-I)
      END-PERFORM
```

All rights reserved to D.Maymone

איפוס שורות

- אם כל שורה מכילה הרבה עמודות, עדיף לאפס את השורה הראשונה ולהעתיק אותה לשאר השורות

```
MOVE ZERO TO TB01-A(1)
             TB01-B(1)
             TB01-D(1)
MOVE SPACES TO TB01-C(1)
                 TB01-E(1)
PERFORM VARYING TB01-I FROM 2 BY 1
            UNTIL TB01-I > 100
   MOVE TB01-BRANCH (1) TO
             TB01-BRANCH (TB01-I)
END-PERFORM
```

All rights reserved to D.Maymone

איפוס טבלאות בינריות

```
01   TB02-ITEMS.
  03 TB02-ITEM      OCCURS 1000
                    INDEXED BY TB02-I.
     05   TB01-A   PIC S9(9) BINARY.
     05   TB01-B   PIC S9(4) BINARY.

 MOVE LOW-VALUE TO TB02-ITEMS
```

All rights reserved to D.Maymone

איפוס ע"י מצביע

- נתון קובץ ממוין ע"פי מספר סניף ומספר לקוח.
- מספר הסניפים כ-4,000.
- בכל סניף עד 500 לקוחות
- לכל לקוח יש מספר רב של תנועות, אותם יש לסכם.
- בסוף כל סניף יש להדפיס טבלה מרוכזת של סכומים שנצברו עבור כל לקוח
- כיצד מאפסים את הטבלה?

All rights reserved to D.Maymone

איפוס מונים

```
01   CN01-COUNTERS.
     03 CN01                 PIC … VALUE 0.
     03 CN02                 PIC … VALUE 0.
     03 CN03                 PIC … VALUE 0.
        ……
01   CN01-INIT                  PIC X(250).
```

- בתחילת התוכנית :

```
MOVE CN01-COUNTERS TO CN01-INIT
```

- בנקודת האיפוס :

```
MOVE CN01-INIT TO CN01-COUNTERS
```

All rights reserved to D.Maymone

איפוס ע"י שטח קבוע

- שמור את הטבלה המאותחלת בשטח אחר והשתמש בו לאתחולים הבאים של אותה טבלה.

```
01   TB03-CUSTOMERS.
      03 TB03-CUSTOMER        OCCURS 15.
           05 TB03-A …
           05 TB03-B …
           05 TB03-C …
……
01   TB03-INIT                    PIC X(300).
```

- מאפסים את TB03 בשיטה כלשהי ושומרים אותה

```
MOVE TB03-CUSTOMERS TO TB03-INIT
```

- בכל איפוס :

```
MOVE TB03-INIT TO TB03-CUSTOMERS
```

All rights reserved to D.Maymone

שיטת המריחה

```
01  TB01-BRANCHES-INIT.
        03 FILLER                PIC 9(5) VALUE 0.
        03 FILLER                PIC 9(9) PACKED-DECIMAL VALUE 0.
        03 FILLER                PIC X(20) VALUE SPACE.
        03 FILLER                PIC S9(9) BINARY VALUE 0.
        03 FILLER                PIC X(5) VALUE SPACES.
        03 TB01-BRANCHES.
           05 TB01-BRANCH        OCCURS 10.
              07 TB01-A          PIC 9(5).
              07 TB01-B          PIC 9(9) PACKED-DECIMAL.
              07 TB01-C          PIC X(20).
              07 TB01-D          PIC S9(9) BINARY.
              07 TB01-E          PIC X(5).
```

MOVE TB01-BRANCHES-INIT TO TB01-BRANCHES

All rights reserved to D.Maymone

שיטות חיפוש

- חיפוש סדרתי
- השיפור של Knuth
- חיפוש בינרי
- אינדקס מלא
- אינדקס חלקי
- עקרון Paretto – 20-80
- Hashing

All rights reserved to D.Maymone

חיפוש סדרתי

- חיפוש סדרתי רגיל
- השיפור של Knuth
- חיפוש סדרתי בטבלאות ממוינות

All rights reserved to D.Maymone

Regular Sequential Search(C, C++, Java, C#)

```
for (i = 0; i < N; i++)
{
     if (a[i] == X)
     {
             printf ("Found in %d\n", i + 1);
             break;
     }
}
if (i >= N)
     printf ("Not Found\n");
```

All rights reserved to D.Maymone

Regular Sequential Search – COBOL

```
SET TB01-I TO 1
SEARCH TB01-ENTRY
     AT END
             DISPLAY 'Entry not found'
     WHEN TB01-ENTRY(TB01-I) = WS01-X
             SET WS02-I TO TB01-I
             DISPLAY 'Entry found in ' WS02-I
END-SEARCH
```

All rights reserved to D.Maymone

Knuth Improvement(C, C++, Java, C#)

```
a[N] = X;
for (i = 0; a[i] != X; i++);
if (i < N)
    printf ("Found in %d\n", i + 1);
else
    printf ("Not Found\n");
```

All rights reserved to D.Maymone

Knuth Improvement – COBOL

```
MOVE WS01-X TO TB01-ENTRY(CO01-NUMBER-OF-ENTRIES)
PERFORM VARYING TB01-I
            FROM 1
            BY 1
            UNTIL TB01-ENTRY(TB01-I) = WS01-X
END-PERFORM
SET WS01-I TO TB01-I
IF WS01-I = CO01-NUMBER-OF-ENTRIES
     DISPLAY ‘Entry not found’
ELSE
     DISLPLAY ‘Entry found in ‘ WS01-I
END-IF
```

All rights reserved to D.Maymone

Sequential Search in Sorted Arrays(C, C++, Java, C#)

```
a[N] = MAX_VALUE;
for (i = 0; a[i] < X; i++);
    if (a[i] == X)
            printf ("Found in %d\n", i + 1);
    else
            printf ("Not Found\n");
```

All rights reserved to D.Maymone

Sequential Search in Sorted Arrays – COBOL

```
MOVE CO01-MAX-VALUE TO TB01-ENTRY(CO01-NUMBER-OF-ENTRIES)
PERFORM VARYING TB01-I
            FROM 1
            BY 1
            UNTIL TB01-ENTRY(TB01-I) >= WS01-X
END-PERFORM
IF TB01-ENTRY(TB01-I) = WS01-X
    SET WS01-I TO TB01-I
    DISPLAY 'Entry found in ' WS01-I
ELSE
    DISPLAY 'Entry not found'
END-IF
```

All rights reserved to D.Maymone

חיפוש בינרי

- השוואה: חיפוש בינרי וחיפוש סדרתי
- תרשים חיפוש בינרי
- קוד חיפוש בינרי
- חיפוש בינרי וטבלאות גדולות
- חיפוש בינרי וקבצי תנועות גדולים

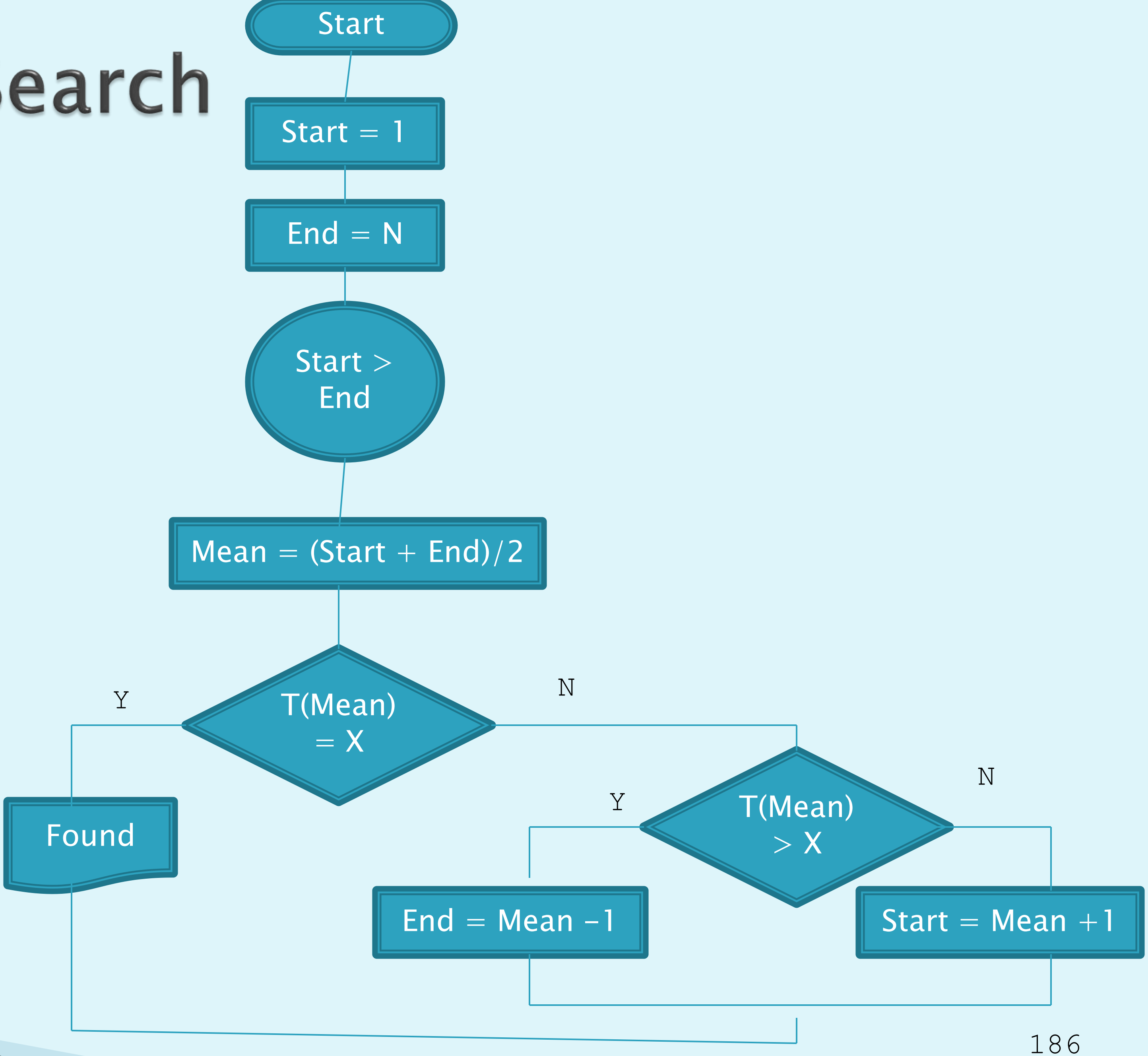
All rights reserved to D.Maymone

Binary Search and Sequential Search

Number of Entries	Sequential	Binary
32	16	5
64	32	6
128	64	7
256	128	8
512	256	9
1024	512	10
2048	1024	11
4096	2048	12

All rights reserved to D.Maymone

Binary Search



All rights reserved to D.Maymone

Binary Search Code (C, C++, Java, C#)

```
start = 0;
end = N - 1;
while (start <= end)
{
      mean = (start + end) / 2;
      current = a[mean];
      if (current == X)
      {
            printf ("Found in %d\n", mean + 1);
            break;
      }
      if (current > X)
            end = mean - 1;
      else
            start = mean + 1;
}
if (start > end)
      printf ("Not Found\n");
```

All rights reserved to D.Maymone

Binary Search Code – COBOL

```
*   See complete solution in BINSRCH.CBL
    MOVE 1 TO WS02-START
    MOVE CO01-NUMBER-OF-ENTRIES TO WS03-END
    PERFORM UNTIL WS02-START > WS03-END OR SW01-ENTRY-FOUND
         COMPUTE WS04-MEAN =
                        FUNCTION MEAN (WS02-START, WS03-END)
         IF TB01-DATE(WS04-MEAN) = WS01-DATE
                SET SW01-ENTRY-FOUND TO TRUE
                EXIT
         END-IF
         IF TB01-DATE(WS04-MEAN) > WS01-DATE
                COMPUTE WS03-END = WS04-MEAN - 1
         ELSE
                COMPUTE WS02-START = WS04-MEAN + 1
         END-IF
    END-PERFORM
    IF NOT SW01-ENTRY-FOUND
         MOVE WS03-END TO WS04-MEAN
    END-IF.
```

All rights reserved to D.Maymone

Binary Search and Large Arrays

- נתון מערך של 1,000 עובדים
- לכל עובד:
  - מספר עובד (9 ספרות)
  - פרטים (200 בתים)

- המערך ממוין לפי מספר עובד
- הגדר את הטבלה לצורך חיפוש בינרי

All rights reserved to D.Maymone

פתרון א'

```
01   TB01-EMPLOYEES.
     03 TB01-EMPLOYEE              OCCURS 1000
                                   INDEXED BY TB01-I
                                   ASCENDING KEY TB01-ID.
          05 TB01-ID               PIC 9(9).
          05 TB01-DETAILS          PIC X(200).
```

- סכנה של PAGING גבוה

All rights reserved to D.Maymone

הפתרון הנכון

```
01  TB01-EMPLOYEES.
    03 TB01-ID                    OCCURS 1000
                                  INDEXED BY TB01-I
                                  ASCENDING KEY TB01-ID
                                  PIC 9(9).
    03 TB01-DETAILS               OCCURS 1000
                                  INDEXED BY TB01-J
                                  PIC X(200).
```

All rights reserved to D.Maymone

חיפוש בינרי וקבצי תנועות גדולים

- נתון קובץ תנועות של 250,000 רשומות
- נתונה טבלה לא ממוינת של 300 ספקים
- בדוק אם הספק בכל אחת מהתנועות נמצא בטבלת הספקים
- האם יש למיין את הטבלה?

All rights reserved to D.Maymone

מאפייני החיפוש הבינרי

- מספר החיפושים = log_2N
- המערך חייב להיות ממויין לפי סדר עולה/יורד
- מומלץ במערכים עם לפחות 50 כניסות
- במערכים גדולים יש להפריד בין המפתח והתוכן
- ההשקעה במיון היא בערך N^2 פעולות

All rights reserved to D.Maymone

בעיית סוגי החשבונות

- נתונה טבלה של 200 סוגי חשבונות
  - סוג חשבון - 3 ספרות
  - פרטים – 100 תווים
- קרא סוג חשבון והצג את תאורו.

All rights reserved to D.Maymone

Full Index

```
01  TB01-ACCOUNTS-TYPES.
    03 TB01-ACCOUNT-DETAILS   OCCURS 200
                              INDEXED BY TB01-I
                              PIC X(100).
     01  TB01-POINTER         OCCURS 1000
                              PIC S9(4) COMP.
```

All rights reserved to D.Maymone

קטע הבניה

```
ADD 1 TO I
IF I > MAX-NUMBER-OF-ACCOUNT-TYPES
  PERFORM X-ERROR
END-IF
MOVE I01-ACCOUNT-DETAILS TO
                        TB01-ACCOUNT-DETAILS(I)
MOVE I TO TB01-POINTER(I01-ACCOUNT-TYPE)
```

All rights reserved to D.Maymone

אינדקס חלקי

- המדינות מחולקות ל- 6 יבשות (יש כ-200 מדינות)
- כל יבשת מאופיינת ע"י קוד 1-6
- בכל יבשת, קוד המדינה מופיע בסדר רץ ורציף 01-50 (בהתאם למספר המדינות ביבשת)
- נתון מערך של מדינות במבנה הבא:
  - קוד יבשת : 9
  - קוד מדינה : 99
  - פרטים – X(20)
- מצא את השיטה היעילה ביותר לאיתור מדינה רצויה

All rights reserved to D.Maymone

פתרונות

סוג חיפוש	עלות
סדרתי	100
בינרי	8
אינדקס מלא	זכרון

- אינדקס חלקי שמצביע על הכניסה הראשונה של כל יבשת.

1	101
2	102
3	103
	...
31	201
	202
	...
66	301

0	
30	
65	
...	

מספר המדינות עד היבשה הזו

All rights reserved to D.Maymone

Hashing

- נתונה טבלה של 1,000 כניסות
- בחיפוש סדרתי
  - אם כל הערכים הדרושים נמצאים אז N/2 ז"א 500 איטרציות
  - אם 50% מהערכים נמצאים אז 0.75N ז"א 750 איטרציות
- בחיפוש סדרתי ממוין - N/2 ז"א 500 איטרציות
- בחיפוש בינרי - LogN ז"א בערך 10 איטרציות.
- במקרים מסוימים יש צורך לשפר ולהגיע לפחות איטרציות

All rights reserved to D.Maymone

Hashing

- יש למצוא M ראשוני >= 1.2N
- בניית טבלה בגודל M כניסות
- הכנסת הערכים הדרושים לטבלה לפי ה- Modulo
- טיפול בכפולים
- חיפוש ערך בטבלה

All rights reserved to D.Maymone

דוגמה - Hashing

1	17
2	51
3	50
4	30
5	55
6	209
7	
8	7
9	
10	
11	197
12	215
13	180
14	200
15	30
16	202
17	152

- 197
- 215
- 55
- 17
- 51
- 200
- 202
- 152
- 209
- 180
- 7
- 50
- 30

All rights reserved to D.Maymone

טיפול בכפולים

- בטבלה עצמה
- ברשימה נפרדת רציפה
- ברשימה מקושרת

All rights reserved to D.Maymone

מערכים גדולים מאוד

- נתונים 170,000 ספקים בקובץ /Indexed
- מספר ספק : (6)9
- קרא קובץ תנועות גדולות (כ- 2 מיליון) לא ממוין ובדוק בכל תנועה האם הספק קיים.

All rights reserved to D.Maymone

שטחי עבודה

```
01   TB01-SUPPLIERS       VALUE LOW-VALUE.
     03 TB01-SUPPLIER     OCCURS 125000
                          PIC X.
01   TB02-BITS            VALUE
                    '128064032016008004002001'
     03 TB02-BIT          OCCURS 8
                          PIC 999.
*------------------------
* WSXX - GENERAL PURPOSE
*------------------------
 01   WS01-NUMBER         PIC S9(4) COMP VALUE 0.
 01   WS01-NUMBER-X       REDEFINES WS01-NUMBER.
      03   WS01-LEFT      PIC X.
      03   WS01-RIGHT     PIC X.
```

All rights reserved to D.Maymone

בניית המערך

```
DIVIDE I01-SUPPLIER-NO BY 8
          GIVING WS02-RESULT
          REMAINDER WS02-REMAINDER
IF WS02-REMAINDER = 0
   MOVE 8 TO WS02-REMAINDER
ELSE
   ADD 1 TO WS02-RESULT
END-IF
MOVE TB01-SUPPLIER(WS02-RESULT) TO WS01-RIGHT
ADD TB02-BIT(WS02-REMAINDER) TO WS01-NUMBER
MOVE WS01-RIGHT TO TB01-SUPPLIER(WS02-RESULT)
```

All rights reserved to D.Maymone

חיפוש במערך

```
 DIVIDE WS03-SUPPLIER-NO BY 8
           GIVING WS02-RESULT
           REMAINDER WS02-REMAINDER
 IF WS02-REMAINDER = 0
      MOVE 8 TO WS02-REMAINDER
 ELSE
      ADD 1 TO WS02-RESULT
 END-IF
 MOVE TB01-SUPPLIER(WS02-RESULT) TO WS01-RIGHT
*  Shift Right
 COMPUTE WS01-NUMBER = WS01-NUMBER / TB02-BIT(WS02-REMAINDER)
 COMPUTE WS02-REMAINDER = FUNCTION MOD(WS01-NUMBER, 2)
 IF WS02-REMAINDER = 0
      DISPLAY 'NOT FOUND'
 ELSE
      DISPLAY 'FOUND'
 END-IF
```

All rights reserved to D.Maymone

שיטות מיון

- מיון בועות
- מיון בועות משופר
- Quick Sort

All rights reserved to D.Maymone

Bubble Sort - מיון בועות(C, C++, Java, C#)

```
void bubbleSort(int a[], int start, int end)
{
        int left, right, temp;
        for (right = end; right > 0; right--)
        {
                for (left = start; left < right; left++)
                {
                        if (a[left] > a[left + 1])
                        {       // Swap
                                temp = a[left];
                                a[left] = a[left + 1];
                                a[left + 1] = temp;
                        }
                }
        }
}
```

All rights reserved to D.Maymone

מיון בועות משופר (C, C++, Java, C#)

```
void bubbleSort(int a[], int start, int end)
{
	int left, right, temp, isSorted = FALSE;
	for (right = end; right > 0 && !isSorted; right--)
	{
		isSorted = TRUE;
		for (left = start; left < right; left++)
		{
			if (a[left] > a[left + 1])
			{		// Swap
				temp = a[left];
				a[left] = a[left + 1];
				a[left + 1] = temp;
				isSorted = FALSE;
			}
		}
	}
}
```

All rights reserved to D.Maymone

מיון בועות משופר (COBOL)

```
SET SW01-NOT-SORTED TO TRUE
PERFORM VARYING TB01-I-LAST
          FROM CO01-NUMBERS
          BY -1
          UNTIL TB01-I-LAST < 2 OR
              SW01-SORTED
      SET SW01-SORTED TO TRUE
      PERFORM VARYING TB01-I
              FROM 1
              BY 1
              UNTIL TB01-I > TB01-I-LAST - 1
          IF TB01-NUMBER(TB01-I) > TB01-NUMBER(TB01-I + 1)
             MOVE TB01-NUMBER(TB01-I) TO WS01-TEMP-NUMBER
              MOVE TB01-NUMBER(TB01-I+ 1) TO
                              TB01-NUMBER(TB01-I)
              MOVE WS01-TEMP-NUMBER  TO TB01-NUMBER(TB01-I + 1)
              SET  SW01-NOT-SORTED   TO TRUE
          END-IF
  END-PERFORM.
```

All rights reserved to D.Maymone

QuickSort(C, C++, Java, C#)

```
void quickSort(int a[], int start, int end)
{
   int left, right, current, temp, mean;
   if (start < end)
   {
         right = end;
         left = start;
         mean = (left + right)/2;
         current = a[mean];
```

All rights reserved to D.Maymone

QuickSort (2) (C, C++, Java, C#)

```
        do
        {
                while (a[left] < current)
                        left ++;
                while (a[right] > current)
                        right --;
                if (left <= right)
                {       // Swap
                        temp = a[left];
                        a[left] = a[right];
                        a[right] = temp;
                        right --;
                        left ++;
                }
        } while (left <= right);
                quickSort(a, start, right);
                quickSort(a, left, end);
        }
}
```

All rights reserved to D.Maymone

QuickSort VB

```
Sub quickSort(ByVal a() As Integer, ByVal vStart As Integer,
  ByVal vEnd As Integer)
        Dim left, right, current, temp, mean As Integer
        If vStart < vEnd Then
            right = vEnd
            left = vStart
            mean = (left + right) / 2
            current = a(mean)
```

All rights reserved to D.Maymone

QuickSort VB (2)

```
        Do
            Do While a(left) < current
                left += 1
            Loop
            Do While (a(right) > current)
                right -= 1
            Loop
            If left <= right Then
                ' Swap
                temp = a(left)
                a(left) = a(right)
                a(right) = temp
                right -= 1
                left += 1
            End If
        Loop While (left <= right)
        quickSort(a, vStart, right)
        quickSort(a, left, vEnd)
    End If
End Sub
```

All rights reserved to D.Maymone

QuickSort (3)

18	3	3	3	3	3	3
51	51	6	6	6	6	6
17	17	17	17	17	17	17
53	53	18	18	18	18	18
6	6	51	33	33	33	33
52	52	52	52	36	36	36
33	33	33	51	51	44	44
44	44	44	44	44	51	51
3	18	53	53	53	53	52
36	36	36	36	52	52	53

All rights reserved to D.Maymone

From Unstructured to Structured

- Loops
- Mana-Ashcroft

All rights reserved to D.Maymone

Loops – 1

```
  IF P GO TO L1.
  PERFORM C.
L5.
  IF Q GO TO L-EXIT.
L1.
  PERFORM D
  GO TO L5.
L-EXIT.
```

- Convert to Structured COBOL.

All rights reserved to D.Maymone

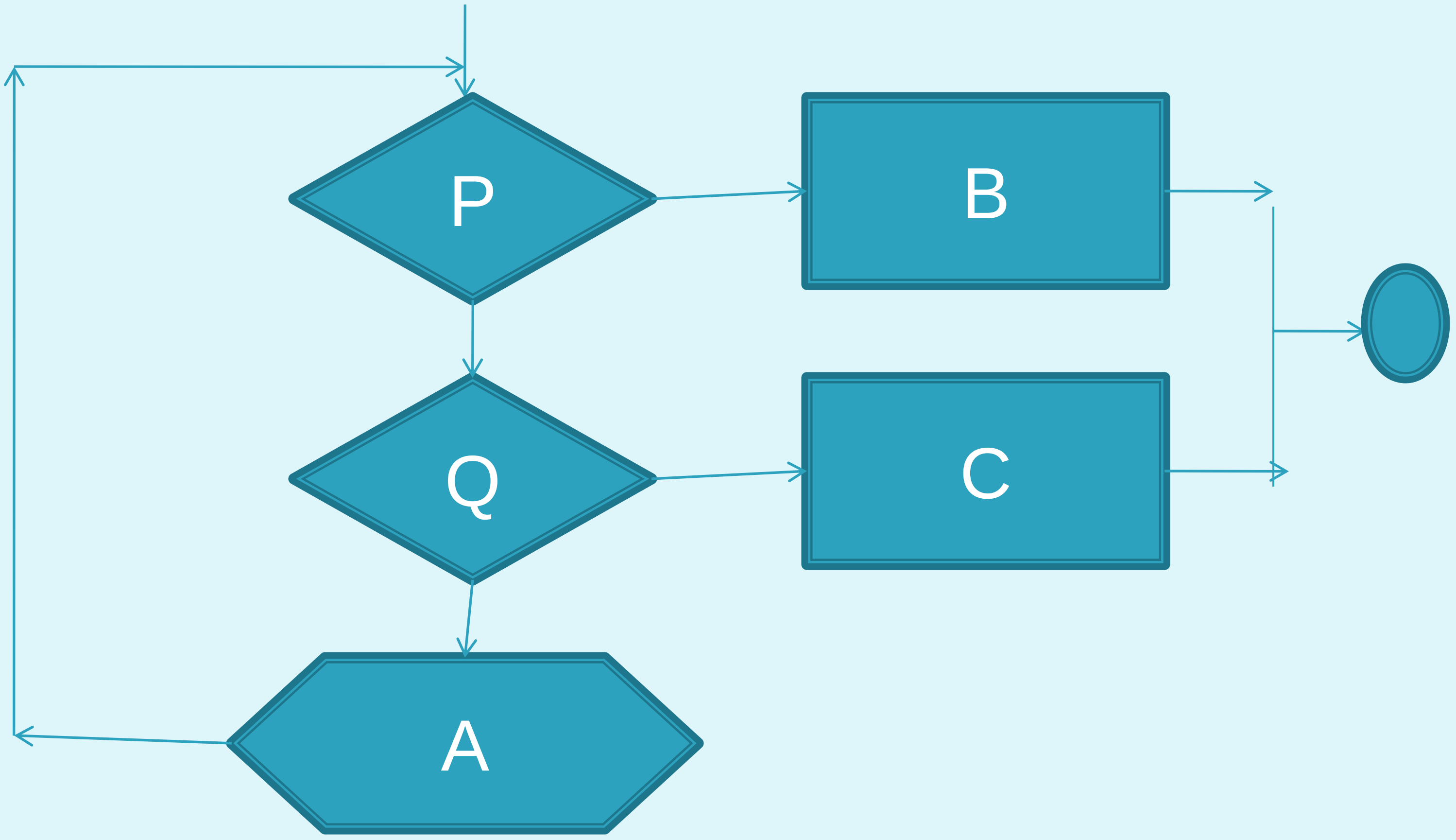
Loops – 2

```
L1.
   IF P GO TO LB.
   IF Q GO TO LC.
   PERFORM A
   GO TO L1.
LB.
   ADD 1 TO X
   MOVE 5 TO Y
   GO TO L-EXIT.
LC.
   COMPUTE A = B + 5
   MOVE 0 TO X Y Z
   ADD 1 TO N.
L-EXIT.
```

- Convert to Structured COBOL.

All rights reserved to D.Maymone

Loops – 2



All rights reserved to D.Maymone

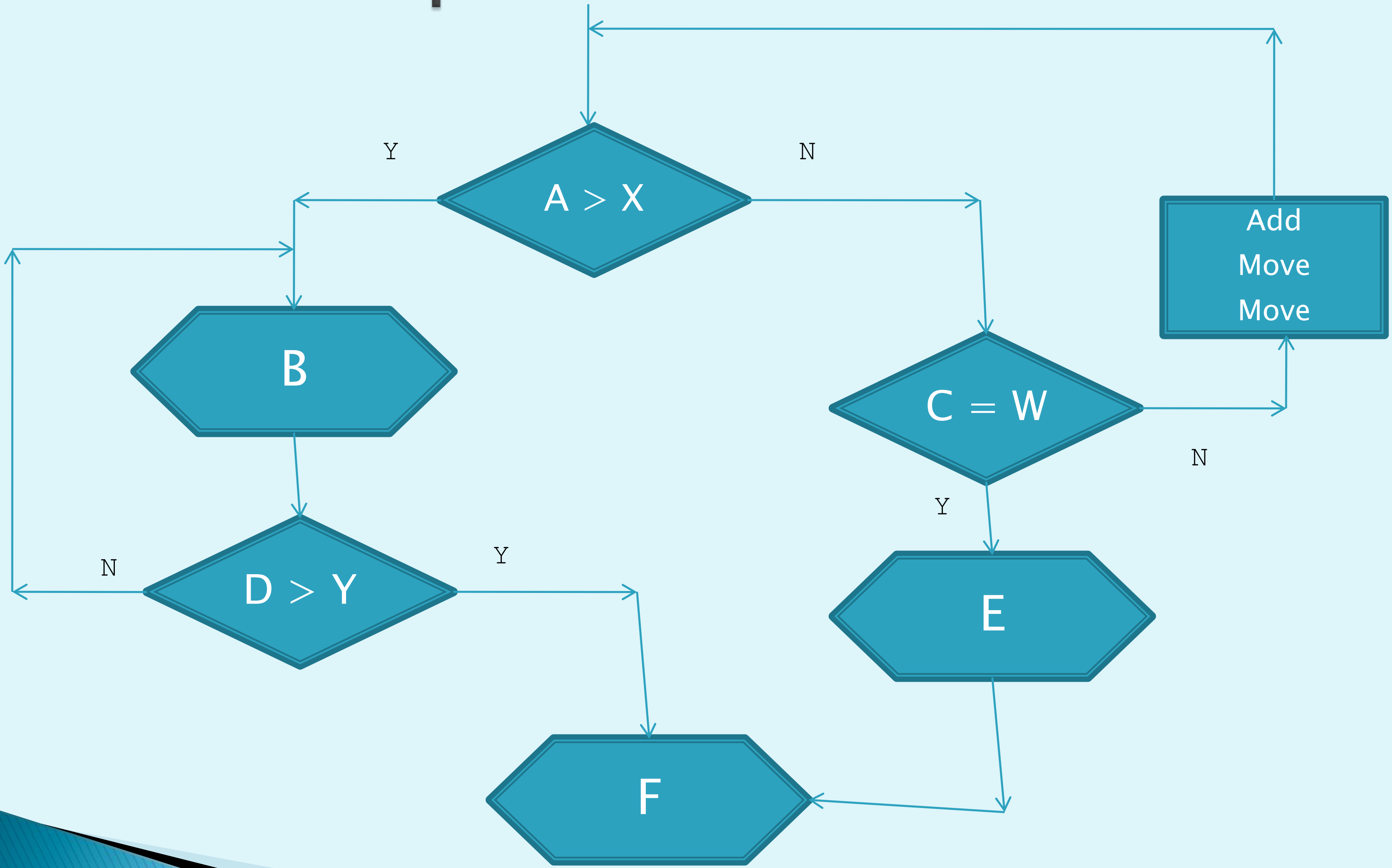
Loops – 3

```
P1.
   IF A NOT > X
         GO TO L1.
L2.
   PERFORM B
   IF D NOT > Y
         GO TO L2.
   GO TO L-SOF.
L1.
   IF C NOT = W
         ADD 1 TO C
         MOVE 0 TO N
         MOVE W TO X
         GO TO P1.
   PERFORM E.
L-SOF.
   PERFORM F.
```

- Convert to Structured COBOL.

All rights reserved to D.Maymone

Loops – 2 – FlowChart



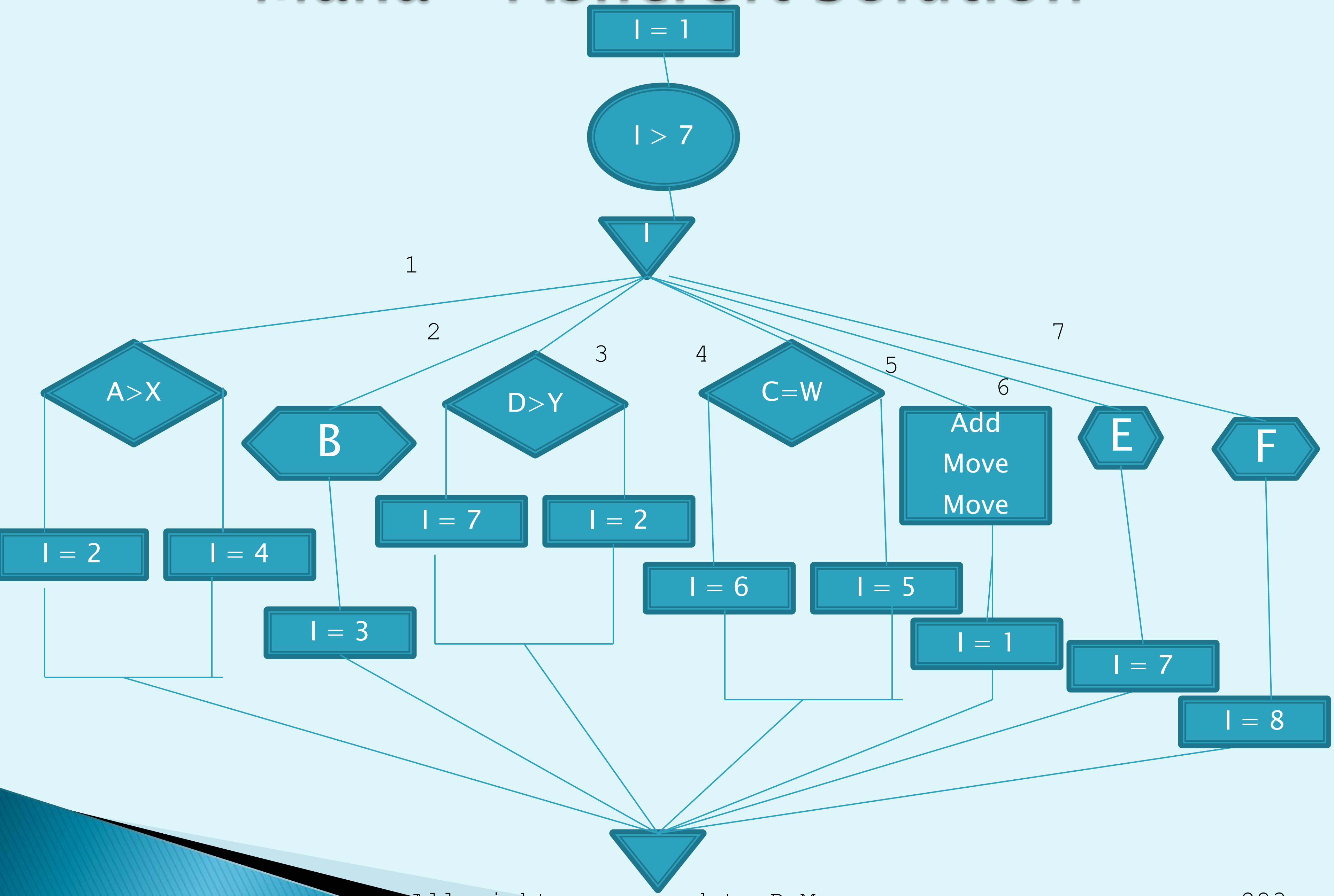
All rights reserved to D.Maymone

Mana Ashcroft

- תן לכל תהליך ולכל תנאי מספר חד ערכי בתחום N–1
- בחר משתנה מצב I. I יציין בכל שלב מהו התהליך הבא
- קבע ערך התחלתי ל- I בהתאם לדרישה
- תרגם את תרשים הזרימה לתהליך שמתבצע עד ש - N > I
- בתוך התהליך בנה צומת שמתפצלת ל- N הפעילויות של הבעיה
- לכל פעילות קבע את ערכו של I הבא

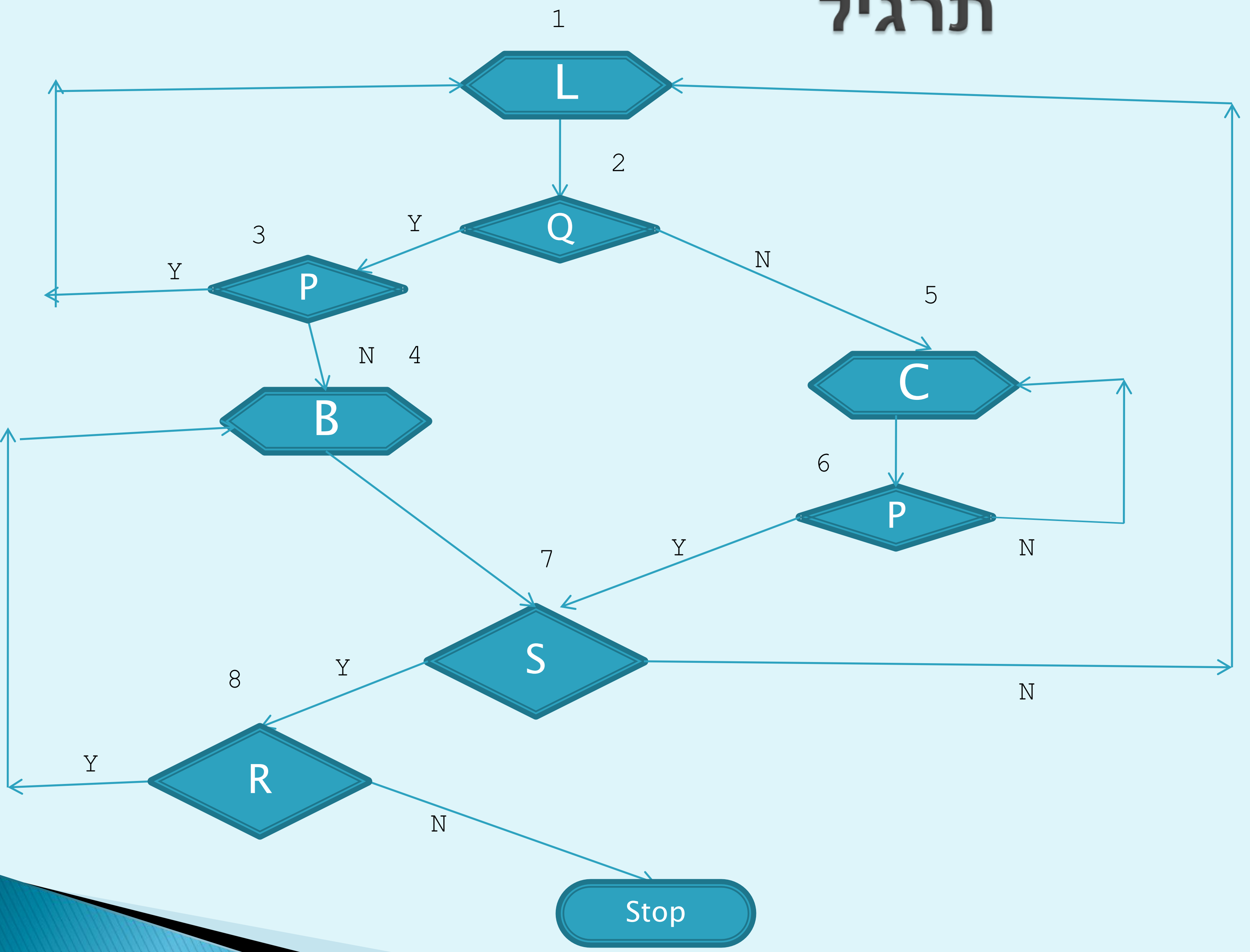
All rights reserved to D.Maymone

Mana – Ashcroft Solution



All rights reserved to D.Maymone

תרגיל



All rights reserved to D.Maymone

Performance

- Structured Programming
- Factoring
- Constants
- Numeric Variables
- Conversions
- SEARCH

All rights reserved to D.Maymone

Structured Programming

- כתיבה בתכנות מבני מאפשרת ל- Optimizer למצוא פתרונות יותר טובים לאופטמיזציה
- השתמש ב- EVALUATE וב- Inline PERFORM
- אין להשתמש בפקודת Alter
- אין לכתוב Go TO "לאחור"

All rights reserved to D.Maymone

Factoring

- שים לב להבדל בין שני הקטעים הבאים:

```
MOVE ZERO TO TOTAL
PERFORM VARYING I FROM 1 BY 1 UNTIL I = 10
  COMPUTE TOTAL = TOTAL + ITEM(I)
END-PERFORM
COMPUTE TOTAL = TOTAL * DISCOUNT

MOVE ZERO TO TOTAL
PERFORM VARYING I FROM 1 BY 1 UNTIL I = 10
  COMPUTE TOTAL = TOTAL + ITEM(I) * DISCOUNT
END-PERFORM
```


All rights reserved to D.Maymone

Constants

- כדי שה- Optimizer יזהה שמדובר בקבוע יש להגדירו מלכתחילה עם Value ולא לשנות אותו בתוכנית

All rights reserved to D.Maymone

Numeric variables

- עם סימן
- BINARY : ‏S9(4) או S9(9)
- כדאי להגדיר מספרים עם יותר מ-9 ספרות ב- Packed-Decimal
- PACKED-DECIMAL : מספר איזוגי של ספרות
- עדיף עד 15 ספרות כדי לא לגרום למערכת לטעון רוטינות ספריה מיוחדת לכפל וחילוק

All rights reserved to D.Maymone

Conversions

- מנע ככל האפשר ביצוע הסבות מיותרות בין משתנים
- כארשר המשתנים משתתפים בחישוב זהים בגודלם ובאופיים לא מתבצעת בדרך כלל הסבה.

All rights reserved to D.Maymone

Performance – SEARCH

- דרוך את האינדקס לכניסה שלפניה לא יכול להימצא הערך הדרוש.
- הפעל את עקרון Paretto : 20-80

All rights reserved to D.Maymone

Exercise 6-3

```
*----------------------------------------
 B-SUMMARIZE                          SECTION.
*----------------------------------------
 B00.
     PERFORM VARYING TB01-I
             FROM 1
             BY 1
             UNTIL TB01-I > CO01-EMPLOYEES
         MOVE 0 TO WS01-EMPLOYEE-SALARIES
         PERFORM VARYING TB01-J
                 FROM 1
                 BY 1
                 UNTIL TB01-J > CO01-MONTHS
             IF TB01-HIGH-SALARY(TB01-I TB01-J)
                 MOVE 0 TO WS01-EMPLOYEE-SALARIES
                 EXIT PERFORM
             END-IF
             ADD TB01-SALARY(TB01-I TB01-J) TO
                                 WS01-EMPLOYEE-SALARIES
         END-PERFORM
         ADD WS01-EMPLOYEE-SALARIES TO WS02-TOTAL-SALARIES
     END-PERFORM.
 B-EXIT.
     EXIT.
```

Exercise 6-3

```
*----------------------------------------
 B-SUMMARIZE                              SECTION.
*----------------------------------------
 B00.
     PERFORM VARYING TB01-I
             FROM 1
             BY 1
             UNTIL TB01-I > CO01-EMPLOYEES
         MOVE 0 TO WS01-EMPLOYEE-SALARIES
         PERFORM VARYING TB01-J
                 FROM 1
                 BY 1
                 UNTIL TB01-J > CO01-MONTHS
             IF TB01-HIGH-SALARY(TB01-I TB01-J)
                 MOVE 0 TO WS01-EMPLOYEE-SALARIES
                 EXIT PERFORM
             END-IF
             ADD TB01-SALARY(TB01-I TB01-J) TO
                                 WS01-EMPLOYEE-SALARIES
         END-PERFORM
         ADD WS01-EMPLOYEE-SALARIES TO WS02-TOTAL-SALARIES
     END-PERFORM.
 B-EXIT.
     EXIT.
```

Decision Tables – 1

```
*-----------------------------------------------
*   Mult |   8   |   4   |   2   |   1   | Supplement |
*   Seq. |   A   |   C   |   SK  |   G   | Supplement |
*-----------------------------------------------
*    1   |   0   |   0   |   0   |   0   |      0     |
*    2   |   0   |   0   |   0   |   1   |      0     |
*    3   |   0   |   0   |   1   |   0   |    250     |
*    4   |   0   |   0   |   1   |   1   |      0     |
*    5   |   0   |   1   |   0   |   0   |    500     |
*    6   |   0   |   1   |   0   |   1   |      0     |
*    7   |   0   |   1   |   1   |   0   |    500     |
*    8   |   0   |   1   |   1   |   1   |    400     |
*    9   |   1   |   0   |   0   |   0   |      0     |
*   10   |   1   |   0   |   0   |   1   |      0     |
*   11   |   1   |   0   |   1   |   0   |    250     |
*   12   |   1   |   0   |   1   |   1   |    200     |
*   13   |   1   |   1   |   0   |   0   |    500     |
*   14   |   1   |   1   |   0   |   1   |      0     |
*   15   |   1   |   1   |   1   |   0   |    500     |
*   16   |   1   |   1   |   1   |   1   |    200     |
*-----------------------------------------------
     01 TB01-SUPPLEMENTS       VALUE
              '000000250000500000500400000000250200500000500200'.
         03 TB01-SUPPLEMENT            OCCURS 16 PIC 999.
```

Decision Tables – 1 – Method 1

```
MOVE 1 TO I
IF WS01-MALE
    ADD CO01-GENDER-MULTIPLIER TO I
END-IF
IF WS02-ADVANCED-SKILLS
    ADD CO02-SKILLS-MULTIPLIER TO I
END-IF
IF WS03-MANY-CHILDREN
    ADD CO03-CHILDREN-MULTIPLIER TO I
END-IF
IF WS04-OLD-EMPLOYEE
    ADD CO04-AGE-MULTIPLIER TO I
END-IF
DISPLAY '1ST-METHOD - THE SUPPLEMENT IS: '
                      TB01-SUPPLEMENT(I).
```

Decision Tables – 1 – Method 2

```
MOVE WS01-GENDER TO WS05-G
IF WS02-ADVANCED-SKILLS
      MOVE 1 TO WS05-S
ELSE
      MOVE 0 TO WS05-S
END-IF
IF WS03-MANY-CHILDREN
      MOVE 1 TO WS05-C
ELSE
      MOVE 0 TO WS05-C
END-IF
IF WS04-OLD-EMPLOYEE
      MOVE 1 TO WS05-A
ELSE
      MOVE 0 TO WS05-A
END-IF
DISPLAY '2ND METHOD - THE SUPPLEMENT IS: '
                     TB01-SUPPLEMENT(I).
```

Thank you
D.Maymone

All rights reserved to D.Maymone