INIT .
IF OP = 1
DISPLAY "SHOP IS OPEN"
DISPLAY "SHOP IS CLOSED"
GO TO INIT
PERFORM INIT-PRD THROUGH INIT-PRD-FN
GO TO INIT-FN
INIT-PRD .
COMPUTE QT-VEG = FUNCTION RANDOM (1) * 10
COMPUTE QT-MEAT = FUNCTION RANDOM (1) * 10
COMPUTE QT-BREAD = FUNCTION RANDOM (1) * 10
COMPUTE QT-MILK = FUNCTION RANDOM (1) * 10
COMPUTE QT-FRUIT = FUNCTION RANDOM (1) * 10
COMPUTE PR-VEG = FUNCTION RANDOM (1) * 10 + 3
COMPUTE PR-MEAT = FUNCTION RANDOM (1) * 10 + 5
COMPUTE PR-BREAD = FUNCTION RANDOM (1) * 10 + 1
COMPUTE PR-MILK = FUNCTION RANDOM (1) * 10 + 2
COMPUTE PR-FRUIT = FUNCTION RANDOM (1) * 10 + 1
INIT-PRD-FN .
EXIT
INIT-FN .
EXIT
BUY-VEG .
PERFORM ISNEEDED THROUGH ISNEEDED-FN
IF NEED = 1 AND QT-VEG > 0
IF MONEY > PR-VEG AND BAG < MAX-CAP
GO TO BUY-MEAT
ADD 1 TO BAG
GO TO PRINT
COMPUTE MONEY = MONEY - PR-VEG
SUBTRACT 1 FROM QT-VEG
END-IF
END-IF
BUY-VEG-FN .
EXIT
BUY-MEAT .
PERFORM ISNEEDED THROUGH ISNEEDED-FN
IF NEED = 1 AND QT-MEAT > 0
IF MONEY > PR-MEAT AND BAG < MAX-CAP
GO TO BUY-BREAD
ADD 1 TO BAG
GO TO PRINT
COMPUTE MONEY = MONEY - PR-MEAT
SUBTRACT 1 FROM QT-MEAT
END-IF
END-IF
BUY-MEAT-FN .
EXIT
BUY-BREAD .
PERFORM ISNEEDED THROUGH ISNEEDED-FN
IF NEED = 1 AND QT-BREAD > 0
IF MONEY > PR-BREAD AND BAG < MAX-CAP
GO TO BUY-MILK
ADD 1 TO BAG
GO TO PRINT
COMPUTE MONEY = MONEY - PR-BREAD
SUBTRACT 1 FROM QT-BREAD
END-IF
END-IF
BUY-BREAD-FN .
EXIT
BUY-MILK .
PERFORM ISNEEDED THRU ISNEEDED-FN
IF NEED = 1 AND QT-MILK > 0
IF MONEY > PR-MILK AND BAG < MAX-CAP
GO TO BUY-FRUIT
ADD 1 TO BAG
GO TO PRINT
COMPUTE MONEY = MONEY - PR-MILK
SUBTRACT 1 FROM QT-MILK
END-IF
END-IF
BUY-MILK-FN .
EXIT
BUY-FRUIT .
PERFORM ISNEEDED THRU ISNEEDED-FN
ISNEEDED .
IF NEED = 1 AND QT-FRUIT > 0
COMPUTE NEED = FUNCTION RANDOM (1) * 2
IF MONEY > PR-FRUIT AND BAG < MAX-CAP
GO TO CHECK
ISNEEDED-FN .
ADD 1 TO BAG
GO TO PRINT
EXIT
COMPUTE MONEY = MONEY - PR-FRUIT
SUBTRACT 1 FROM QT-FRUIT
END-IF
END-IF
BUY-FRUIT-FN .
EXIT
CHECK .
IF MONEY <= 0 OR BAG >= MAX-CAP
GO TO PRINT
GO TO BUY-VEG
PRINT .
MOVE MONEY TO REST
DISPLAY "REST:" MONEY
DISPLAY "NB OF PRODUCTS:" BAG
FIN .
STOP RUN
sequential next
procedure call
procedure backward call
perform-exit
true
false

test