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

test