**START**

Declare structure inventory.

Declare pointer1, pointer2 as file pointer(s) in inventory.

Declare tempint, tempint2 as integers in inventory.

Declare tempfloat, tempfloat2 as floats in inventory.

Declare output as character pointer in inventory.

Declare barcode as character array of length 15 in inventory.

Declare label as character array of length 50 in inventory.

Declare price as float in inventory.

Declare tempchar as character array of length 100 in inventory.

Declare size as a size holder in inventory.

Declare Locate(), Append(), Display(), Delete(), Search(), Stocks(), DataStore(float price), IncomeCalculator as void functions.

char barcodetest[ ] 🡨 “MGSI”.

Declare choice as character array of length 100.

**main()**

**begin**

Writeln(“MATTHEW’S GROCERY STORE INFORMATION”)

**Locate()**

**Writeln(“INVENTORY OPTIONS”)**

**Writeln(“a = Append/Create inventory”)**

**Writeln(“b = Display inventory”)**

**Writeln(“c = Delete inventory”)**

**Writeln(“d = Search inventory”)**

**Writeln(“e = Calculate stocks”)**

**Writeln(“f = Income Calculator”)**

**Write(“Choice: ”)**

**Readln(choice)**

**int error 🡸 stringlengthof(choice)**

**if(error>1) then**

**Writeln(“Error parsing data. Check input.”)**

**StopProgram**

**else if(choice[0] = ‘a’) then**

**Append()**

**else if(choice[0] = ‘b’) then**

**Display()**

**else if(choice[0] = ‘c’) then**

**Delete()**

**else if(choice[0] = ‘d’) then**

**Search()**

**else if(choice[0] = ‘e’) then**

**Stocks()**

**else if(choice[0] = ‘f’) then**

**IncomeCalculator()**

**else**

**Writeln(“Error parsing data. Check input.”)**

**StopProgram**

**endif**

**Write(“Continue(y):”)**

**Read(choice)**

**error 🡸 stringlengthof(choice)**

**if(error>1) then**

**Writeln(“Wrong input.”)**

**StopProgram**

**endif**

**if(choice[0] = ‘y’ or choice[0] = ‘Y’) then**

**ClearScreen**

**main()**

**endif**

**end**

**Locate()**

**begin**

Declare instance “locate” of structure inventory.

locate.pointer 🡸 database.dat in read binary mode

if(locate.pointer==null) then

Writeln(“There is no file.”)

else

Writeln(“Successfully located file.”)

endif

closefile(locate.pointer)

**end**

**Append()**

**begin**

Declare instance “append” of structure inventory.

Write(“8-Digit Barcode#(MGSI----): ”)

Readln(append.barcode)

append.tempint 🡸 stringlengthof(append.barcode)

if((append.barcode & barcodetest have the same 4 chars)or(append.tempint!=8)) then

Writeln(“Wrong input.”)

StopProgram

endif

**Write(“Name: ”)**

**Readln(append.label)**

**Write(“Price: $”)**

**Readln(append.price)**

**if(append.price=0 or append.price<0) then**

**Writeln(“Wrong input.”)**

**StopProgram**

**endif**

**append.pointer 🡸 database.dat in append binary mode**

**WriteFile(append.barcode)**

**Write(“Copied [”, append.barcode, “]”)**

**WriteFile(append.price)**

**Write(“, [”, append.price, “]”)**

**WriteFile(append.label)**

**Write(“ and [”, append.label, “]”)**

**closefile(append.pointer)**

**DataStore(append.price)**

**end**

**Display()**

**begin**

Declare instance “display” of structure inventory.

display.pointer 🡸 database.dat in read binary mode

if(display.pointer=NULL) then

Writeln(“There is no file.”)

else

display.size 🡸 sizeof(display.pointer)

display.output[display.size] 🡸 0

Writeln(“Format: ( Barcode Number / Price($) / Label”)

Writeln(display.output)

endif

closefile(display.pointer)

**end**

**Delete()**

**begin**

delete(database.dat)

delete(price.dat)

Writeln(“Successfully deleted file(s).”)

**end**

**Search()**

**begin**

Declare instance “search” of structure inventory.

search.pointer 🡸 database.dat in read binary mode

if(search.pointer=NULL) then

Writeln(“There is no file.”)

else

Write(“8-Digit Barcode#(MGSI----):”)

Readln(search.barcode)

search.tempint 🡸 stringlengthof(search.barcode)

if((search.barcode, barcodetest has same 1st 4 chars)or(search.temp!=8) then

Writeln(“Wrong input.”)

StopProgram

endif

while(endof[search.pointer]is not reached) do

search.tempchar 🡸 ReadLine(search.pointer)

search.output 🡸 stringsearch(search.tempchar, search.barcode)

if(search.output=NULL) then

Writeln(“Not found.”)

else

Writeln(“Found record.”)

Writeln(search.tempchar);

return

endif

endwhile

endif

closefile(search.pointer)

**end**

**DataStore(float price)**

**begin**

Declare instance “data” of structure inventory.

data.pointer 🡸 price.dat in read binary mode

if(data.pointer=NULL) then

closefile(data.pointer)

data.pointer 🡸 price.dat in write binary mode

WriteFile(price, 1)

closefile(data.pointer)

else

ReadFile(data.tempfloat, data.tempint)

price 🡸 price + data.tempfloat

data.tempint 🡸 data.tempint + 1

closefile(data.pointer)

data.pointer 🡸 price.dat in write binary mode

WriteFile(price, data.tempint)

closefile(data.pointer)

endif

**end**

**Stocks()**

**begin**

Declare instance “stocks” of structure inventory

Declare budget, income as floats

Write(“Current Budget: $”)

Readln(budget)

**if(budget=0) then**

**Writeln(“Wrong input.”)**

**StopProgram**

**endif**

**Write(“Total Income: $”)**

**Readln(income)**

**if(income=0) then**

**Writeln(“Wrong input.”)**

**StopProgram**

**stocks.pointer 🡸 price.dat in read binary mode**

**ReadFile(stocks.price, stocks.tempint)**

**Writeln(“Total items: ”, stocks.tempint)**

**Writeln(“Total Inventory Price: ”, stocks.price)**

**closefile(stocks.pointer)**

**printf(“Total Budget: ”, (budget+stocks.price))**

**printf(“Old Budget: ”, (budget-income))**

**end**

**IncomeCalculator()**

**begin**

Declare i, itemssold as integers

Declare current\_price, total\_price as floats

total\_price 🡸 0

Write(“Items Sold: ”)

Readln(itemssold)

if(itemssold=0) then

Writeln(“Wrong input.”)

StopProgram

endif

itemssold 🡸 itemssold + 1

for(i = 1 to itemssolds) do

Write(“Price of item ”, i, “: $”)

Readln(current\_price)

if(current\_price=0) then

Writeln(“Wrong input.”)

StopProgram

endif

total\_price 🡸 total\_price + current\_price

Writeln(“”)

**endfor**

**Writeln(“Total Income: $”, total\_price)**

**end**

**STOP**