2.16 LAB Introduction House real estate summary

```
integer currPrice
integer lastMonthPrice
float estMortgage

currPrice = Get next input
lastMonthPrice = Get next input

Put "This house is $" to output
Put currPrice to output
Put ". The change is $" to output
Put currPrice - lastMonthPrice to output
Put " since last month.\n" to output

estMortgage = (currPrice * 0.045) / 12
Put "The estimated monthly mortgage is $" to output
Put estMortgage to output with 2 decimal places
Put "." to output
```

2.17 LAB Introduction Caffeine levels

```
float caffeineAmt

caffeineAmt = Get next input

caffeineAmt = caffeineAmt / 2
Put "After 6 hours: " to output
Put caffeineAmt to output
Put " mg\n" to output

caffeineAmt = caffeineAmt / 2
Put "After 12 hours: " to output
Put caffeineAmt to output
Put " mg\n" to output

caffeineAmt = caffeineAmt / 2
Put "After 18 hours: " to output
Put caffeineAmt to output
Put " mg" to output
```

2.18 LAB VariablesAssignments Divide by x

```
integer userNum
integer x

userNum = Get next input
x = Get next input

userNum = userNum / x
Put userNum to output
Put " " to output

userNum = userNum / x
Put userNum to output
Put " " to output

userNum = userNum / x
Put userNum to output
Put " " to output

userNum = userNum/x
Put userNum to output
```

2.19 LAB VariablesAssignments Driving costs

```
float milesPerGallon
float dollarsPerGallon
float dollarsPerMile
float result

milesPerGallon = Get next input
dollarsPerGallon = Get next input
dollarsPerMile = dollarsPerGallon / milesPerGallon

result = 10 * dollarsPerMile
Put result to output with 2 decimal places
Put " " to output

result = 50 * dollarsPerMile
Put result to output with 2 decimal places
Put " " to output

result = 400 * dollarsPerMile
Put result to output with 2 decimal places
```

2.20 LAB VariablesAssignments Simple statistics

```
integer input1
integer input2
integer input3
integer iAvg
integer iProduct
float fAvg
float fProduct

input1 = Get next input
input2 = Get next input
input3 = Get next input

iAvg = (input1 + input2 + input3) / 3
iProduct = input1 * input2 * input3

Put iAvg to output
Put " " to output
Put iProduct to output
Put "\n" to output

fAvg = (input1 + input2 + input3)
fAvg = fAvg / 3
fProduct = input1 * input2 * input3

Put fAvg to output with 5 decimal places
Put " " to output
Put fProduct to output with 5 decimal places
```

2.21 LAB VariablesAssignments Using math functions

```
float x
float y
float z

x = Get next input
y = Get next input
z = Get next input

Put RaiseToPower(x, y) to output with 5 decimal places
Put " " to output
Put RaiseToPower(x, RaiseToPower(y, z)) to output with 5 decimal places
Put " " to output
Put AbsoluteValue(x) to output with 5 decimal places
Put " " to output
Put SquareRoot(RaiseToPower(x * y, z)) to output with 5 decimal places
```

2.22 LAB VariablesAssignments Musical note frequencies

```
float startingFreq
float r

startingFreq = Get next input
r = RaiseToPower(2, 1.0 / 12.0)

Put startingFreq to output with 5 decimal places
Put " " to output
Put RaiseToPower(r, 1) * startingFreq to output with 5 decimal places
Put " " to output
Put RaiseToPower(r, 2) * startingFreq to output with 5 decimal places
Put " " to output
Put RaiseToPower(r, 3) * startingFreq to output with 5 decimal places
Put " " to output
Put RaiseToPower(r, 4) * startingFreq to output with 5 decimal places
```

2.23 LAB VariablesAssignments Phone number breakdown

```
integer phoneNum

phoneNum = Get next input

Put phoneNum / 10000 to output
Put "-" to output
Put phoneNum % 10000 to output
```

3.12 LAB Branches Largest number

```
integer num1
integer num2
integer num3

num1 = Get next input
num2 = Get next input
num3 = Get next input

if (num1 >= num2) and (num1 >= num3)
   Put num1 to output
elseif (num2 >= num1) and (num2 >= num3)
   Put num2 to output
else
   Put num3 to output
```

3.13 LAB Branches Remove gray from RGB

```
integer redAmount
integer greenAmount
integer blueAmount
integer minValue

// Get rgb color values
redAmount = Get next input
greenAmount = Get next input
blueAmount = Get next input

// Find minimum value
minValue = redAmount
if greenAmount < minValue
  minValue = greenAmount
if blueAmount < minValue
  minValue = blueAmount

// Subtract minimum value from all colors, thus removing any gray
redAmount = redAmount - minValue
greenAmount = greenAmount - minValue
blueAmount = blueAmount - minValue

// Output new color values with gray removed
Put redAmount to output
Put " " to output
Put greenAmount to output
Put " " to output
Put blueAmount to output
```

3.14 LAB Branches Leap Year

```
integer inputYear
inputYear = Get next input

Put inputYear to output
Put " is " to output

if (inputYear % 4) == 0
  if((inputYear % 100) == 0) and ((inputYear % 400) != 0)
    Put "not " to output
else
  Put "not " to output

Put "a leap year." to output
```

3.15 LAB Branches Interstate highway numbers

```
integer highwayNumber
integer primaryNumber

highwayNumber = Get next input

// Number invalid
if ((highwayNumber < 1) or (highwayNumber > 999))
  Put highwayNumber to output
  Put " is not a valid interstate highway number." to output

// Number valid
else
  // Auxiliary highway
  if (highwayNumber > 99)
    Put "The " to output
    Put highwayNumber to output
    Put " is auxiliary" to output
    // Get the primary rightmost 2 digits
    primaryNumber = highwayNumber % 100
    Put ", serving the " to output
    Put primaryNumber to output

  // Must be 1-99 (Primary highway)
  else
    primaryNumber = highwayNumber
    Put "The " to output
    Put primaryNumber to output
    Put " is primary" to output

  // Even is east/west
  if ((primaryNumber % 2) == 0)
    Put ", going east/west." to output

  // Odd is north/south
  else
    Put ", going north/south." to output
```

3.16 LAB Branches Seasons

```
integer inputMonth
integer inputDay

inputMonth = Get next input
inputDay = Get next input

if (inputMonth == 1 and inputDay >= 1 and inputDay <= 31)
  Put "winter" to output

elseif (inputMonth == 2 and inputDay >= 1 and inputDay <= 29)
  Put "winter" to output

elseif (inputMonth == 3)
  if (inputDay >= 1 and inputDay <= 19)
    Put "winter" to output
  elseif (inputDay > 19 and inputDay <= 31)
    Put "spring" to output
  else
    Put "invalid" to output

elseif (inputMonth == 4 and inputDay >= 1 and inputDay <= 30)
  Put "spring" to output

elseif (inputMonth == 5 and inputDay >= 1 and inputDay <= 30)
  Put "spring" to output

elseif (inputMonth == 6)
  if (inputDay >= 1 and inputDay <= 20)
    Put "spring" to output
  elseif (inputDay > 20 and inputDay <= 30)
    Put "summer" to output
  else
    Put "invalid" to output

elseif (inputMonth == 7 and inputDay >= 1 and inputDay <= 31)
  Put "summer" to output

elseif (inputMonth == 8 and inputDay >= 1 and inputDay <= 31)
  Put "summer" to output

elseif (inputMonth == 9)
  if (inputDay >= 1 and inputDay <= 21)
    Put "summer" to output
  elseif (inputDay > 21 and inputDay <= 30)
    Put "autumn" to output
  else
    Put "invalid" to output

elseif (inputMonth == 10 and inputDay >= 1 and inputDay <= 31)
  Put "autumn" to output
```

```
elseif (inputMonth == 11 and inputDay >= 1 and inputDay <= 30)
  Put "autumn" to output

elseif (inputMonth == 12)
  if (inputDay >= 1 and inputDay <= 20)
    Put "autumn" to output
  elseif (inputDay > 20 and inputDay <= 31)
    Put "winter" to output
  else
    Put "invalid" to output

else
  Put "invalid" to output
```

3.17 LAB Branches Exact change

```
integer inputVal
integer numDollars
integer numQuarters
integer numDimes
integer numNickels
integer numPennies

inputVal = Get next input

if (inputVal <= 0)
  Put "no change\n" to output

numDollars = inputVal / 100
inputVal = inputVal - numDollars * 100

numQuarters = inputVal / 25
inputVal = inputVal - numQuarters * 25

numDimes = inputVal / 10
inputVal = inputVal - numDimes * 10

numNickels = inputVal / 5
inputVal = inputVal - numNickels * 5

numPennies = inputVal

if numDollars > 0
  Put numDollars to output
  if numDollars == 1
    Put " dollar\n" to output
  else
    Put " dollars\n" to output

if numQuarters > 0
  Put numQuarters to output
  if numQuarters == 1
    Put " quarter\n" to output
  else
    Put " quarters\n" to output

if numDimes > 0
  Put numDimes to output
  if numDimes == 1
    Put " dime\n" to output
  else
    Put " dimes\n" to output

if numNickels > 0
  Put numNickels to output
  if numNickels == 1
    Put " nickel\n" to output
  else
    Put " nickels\n" to output

if numPennies > 0
  Put numPennies to output
  if numPennies == 1
    Put " penny\n" to output
  else
    Put " pennies\n" to output
```

4.12 LAB Loops Convert to binary

```
integer userNum
userNum = Get next input

while userNum > 0
   Put userNum % 2 to output
   userNum = userNum / 2
```

4.13 LAB Loops Varied amount of input data

```
integer userInt
integer sumInts
integer numInts
integer maxInt

sumInts = 0
numInts = 0
maxInt = -1

userInt = Get next input

while userInt >= 0
  sumInts = sumInts + userInt
  numInts = numInts + 1
  if userInt > maxInt
    maxInt = userInt
  userInt = Get next input
Put sumInts / numInts to output
Put " " to output
Put maxInt to output
```

4.14 LAB Loops Output range with increment of 10

```
integer num1
integer num2
integer numToPrint

num1 = Get next input
num2 = Get next input

if num2 < num1
  Put "Second integer can't be less than the first." to output
else
  for numToPrint = num1; numToPrint <= num2; numToPrint = numToPrint + 10
    Put numToPrint to output
    Put " " to output
```

4.15 LAB Loops Countdown until matching digits

```
integer userNum
integer leftDigit
integer rightDigit

userNum = Get next input
if (userNum < 20) or (userNum > 98)
  Put "Input must be 20-98" to output
else
  leftDigit = 0
  rightDigit = 1
  while leftDigit != rightDigit
    Put userNum to output
    Put " " to output
    rightDigit = userNum % 10
    leftDigit = userNum / 10
    userNum = userNum - 1
```

5.10 LAB Arrays Output values below an amount

```
integer i
integer j
integer array(5) userValues
integer upperThreshold

for i = 0; i < userValues.size; i = i + 1
  userValues[i] = Get next input

upperThreshold = Get next input

for j = 0; j < userValues.size; j = j + 1
  if userValues[j] <= upperThreshold
    Put userValues[j] to output
    Put " " to output

Put "\n" to output
```

5.8 LAB Arrays Output numbers in reverse

```
integer array(10) userInts
integer i

for i = 0; i < userInts.size; i = i + 1
   userInts[i] = Get next input

for i = userInts.size - 1; i >= 0; i = i - 1
   Put userInts[i] to output
   Put " " to output
Put "\n" to output
```

5.9 LAB Arrays Middle item

```
integer array(9) userValues
integer currValue
integer midIndex
integer i

currValue = Get next input

while (currValue >= 0) and (i < 9)
   userValues[i] = currValue
   currValue = Get next input
   i = i + 1

if currValue >= 0
   Put "Too many inputs" to output
else
   midIndex = i / 2
   Put userValues[midIndex] to output
```

6.10 LAB User-Defined Functions Driving cost

```
Function DrivingCost(float drivenMiles, float milesPerGallon, float dollarsPerGallon) returns float costMiles
   costMiles = drivenMiles * (1.0 / milesPerGallon) * dollarsPerGallon

Function Main() returns nothing
   float milesPerGallon
   float dollarsPerGallon

   milesPerGallon = Get next input
   dollarsPerGallon = Get next input

   Put DrivingCost(10.0, milesPerGallon, dollarsPerGallon) to output with 2 decimal places
   Put " " to output
   Put DrivingCost(50.0, milesPerGallon, dollarsPerGallon) to output with 2 decimal places
   Put " " to output
   Put DrivingCost(400.0, milesPerGallon, dollarsPerGallon) to output with 2 decimal places
```

6.11 LAB User-Defined Functions Step counter

```
Function StepsToMiles(integer userSteps) returns float numMiles
   numMiles = userSteps / 2000.0

Function Main() returns nothing
   integer inputSteps

   inputSteps = Get next input

   Put StepsToMiles(inputSteps) to output with 4 decimal places
```

6.12 LAB User-Defined Functions Flip a coin

```
Function PrintCoinFlips(integer numFlips) returns nothing
   integer randVal
   integer i

   for i = 0; i < numFlips; i = i + 1
      randVal = RandomNumber(0, 1)
      if randVal == 0
         Put  "heads" to output
      else
         Put  "tails" to output
      Put "\n" to output

Function Main() returns nothing
   integer numFlips

   SeedRandomNumbers(2)

   numFlips = Get next input
   PrintCoinFlips(numFlips)
```

6.13 LAB User-Defined Functions A jiffy

```
Function SecondsToJiffies(float userSeconds) returns float userJiffies
   userJiffies = userSeconds * 100.0

Function Main() returns nothing
   float userInput

   userInput = Get next input

   Put SecondsToJiffies(userInput) to output
```

6.14 LAB User-Defined Functions Leap year

```
Function OutputLeapYear(integer inputYear) returns nothing
  Put inputYear to output
  Put " is " to output

  if (inputYear % 4) == 0
    if ((inputYear % 100) == 0) and ((inputYear % 400) != 0)
      Put "not " to output
  else
    Put "not " to output

  Put "a leap year." to output

Function Main() returns nothing
  integer inputYear

  inputYear = Get next input

  OutputLeapYear(inputYear)
```

6.15 LAB User-Defined Functions Max and min numbers

```
Function SmallestNumber(integer num1, integer num2, integer num3) returns integer smallestNum
  if (num1 <= num2) and (num1 <= num3)
    smallestNum = num1
  elseif num2 <= num3
    smallestNum = num2
  else
    smallestNum = num3

Function LargestNumber(integer num1, integer num2, integer num3) returns integer largestNum
  if (num1 >= num2) and (num1 >= num3)
    largestNum = num1
  elseif num2 >= num3
    largestNum = num2
  else
    largestNum = num3

Function Main() returns nothing
  integer userInput1
  integer userInput2
  integer userInput3

  userInput1 = Get next input
  userInput2 = Get next input
  userInput3 = Get next input

  Put "largest: " to output
  Put LargestNumber(userInput1, userInput2, userInput3) to output
  Put "\nsmallest: " to output
  Put SmallestNumber(userInput1, userInput2, userInput3) to output
```

6.16 LAB User-Defined Functions Convert to binary

```
Function IntegerToBinary(integer num) returns nothing
  while num > 0
    Put num % 2 to output
    num = num / 2

Function Main() returns nothing
  integer userInput

  userInput = Get next input

  IntegerToBinary(userInput)
```

6.17 LAB User-Defined Functions Output values below an amount

```
Function outputIntsLessThanOrEqualToThreshold(integer array(?) userVals, integer upperThreshold) returns nothing
  integer i
  for i = 0; i < userVals.size; i = i + 1
    if userVals[i] <= upperThreshold
      Put userVals[i] to output
      Put " " to output

Function Main() returns nothing
  integer array(5) userVals
  integer upperThreshold
  integer i

  for i = 0; i < userVals.size; i = i + 1
    userVals[i] = Get next input

  upperThreshold = Get next input

  outputIntsLessThanOrEqualToThreshold(userVals, upperThreshold)
```

6.18 LAB User-Defined Functions Adjust list by normalizing

```
Function getMinimumInt(integer array(?) userVals) returns integer minInt
  integer i
  minInt = userVals[0]
  for i = 0; i < userVals.size; i = i + 1
    if userVals[i] < minInt
      minInt = userVals[i]

Function Main() returns nothing
  integer array(5) userVals
  integer currValue
  integer minValue
  integer i

  for i = 0; i < userVals.size; i = i + 1
    userVals[i] = Get next input

  minValue = getMinimumInt(userVals)

  for i = 0; i < userVals.size; i = i + 1
    Put userVals[i] - minValue to output
    Put " " to output
```

6.8 LAB User-Defined Functions Miles to track laps

```
Function MilesToLaps(float userMiles) returns float userLaps
   userLaps = userMiles / 0.25

Function Main() returns nothing
   float userMiles

   userMiles = Get next input
   Put MilesToLaps(userMiles) to output
```

6.9 LAB User-Defined Functions Max magnitude

```
Function MaxMagnitude(integer x, integer y) returns integer largestVal
  if AbsoluteValue(x) > AbsoluteValue(y)
    largestVal = x
  else
    largestVal = y

Function Main() returns nothing
  integer x
  integer y

  x = Get next input
  y = Get next input

  Put MaxMagnitude(x, y) to output
```