**//CODE BY Lucas Tsetskhladze (0478/22 Paper 2 February/March 2017)**

DECLARE GLOBAL AgeOfStudent[0:649] AS INTEGER

DECLARE GLOBAL SchoolHouse[0:649] AS CHARACTER

DECLARE GLOBAL ReactionTime[0:649] AS FLOAT

CALL Initialize()

CALL Menu()

PROCEDURE Initialize()

DECLARE Counter AS INTEGER

Counter ← 0

FOR Counter is 0 to 649

AgeOfStudent[Counter] <- 0

SchoolHouse[Counter] <- ‘’

ReactionTime[Counter] <- 0.0

ENDFOR

ENDPROCEDURE

PROCEDURE Menu()

DECLARE MenuOption AS INTEGER

MenuOption <- 0

WHILE MenuOption NOT 100 DO

OUTPUT(“Select an option”)

OUTPUT(“1 For Inputting”)

OUTPUT(“2 For Average times”)

OUTPUT(“3 For Record Browsing”)

OUTPUT(“100 for exit)

MenuOption <- INT(INPUT)

IF MenuOption = 1 THEN

CALL Inputting()

ELSE

IF MenuOption = 2 THEN

CALL AverageT()

ELSE

IF MenuOption = 3 THEN

CALL Records()

ELSE

IF MenuOption NOT 100 THEN

OUTPUT(“Choose an actual number”)

ENDIF

ENDIF

ENDIF

ENDIF

ENDWHILE

ENDPROCEDURE

PROCEDURE Inputting()

DECLARE Stop AS BOOLEAN

DECLARE Counter AS INTEGER

DECLARE Temporar AS INTEGER

Stop <- False

Counter ← 0

Temporart ← 0

WHILE Stop = False

OUTPUT(“What is your age?”)

AgeOfStudent[Counter] <- CheckingAge(INT(INPUT))

OUTPUT(“What is your House S For Saturn and M For Mars?”)

SchoolHouse[Counter] <- CheckingHouse(UPPERCASE(CHAR(INPUT)))

OUTPUT(“What was the time in milliseconds ?”)

ReactionTime[Counter] <- CheckingReactionTime(FLOAT(INPUT))

OUTPUT(“Another Student? 0 for YES, 1 for NO “)

Temporart <- INT(INPUT)

WHILE Temporart =! 1 AND Temporart =! 0

OUTPUT(“Please Enter a valid number either 1 For No or 0 For

Yes”)

Temporart <- INT(INPUT)

ENDWHILE

Stop ← BOOL(Temporart)

Counter = Counter + 1

ENDWHILE

ENDPROCEDURE

//TASK 2

PROCEDURE AverageT()

DECLARE Counter AS INTEGER

DECLARE TotalReactMars AS FLOAT

DECLARE TotalNumberMars AS INTEGER

DECLARE TotalReactSaturn AS FLOAT

DECLARE TotalNumberSaturn AS INTEGER

Counter ← 0

TotalReactMars ← 0.0

TotalNumberMars ← 0

TotalReactSaturn ← 0.0

TotalNumberSaturn ← 0

WHILE SchoolHouse =! ‘’

IF Schoolhouse[Counter] = ‘M’

TotalReactMars = TotalReactMars + ReactionTime[Counter]

TotalNumberMars = TotalNumberMars + 1

ELSE

TotalReactSaturn = TotalReactSaturn + ReactionTime[Counter]

TotalNumberSaturn = TotalNumberSaturn + 1

ENDIF

Counter = Counter + 1

ENDWHILE

TotalReactMars = TotalReactMars / TotalNumberMars

TotalReactSaturn = TotalReactSaturn / TotalNumberSaturn

//Output Message For Task 2

OUTPUT(“The Average Time for Mars for ”&STR(TotalNumberMars)&” Students is ”&STR(TotalReactMars)&” Milliseconds. The Average Time for Saturn for ”&STR(TotalNumberSaturn)&” Students is ”&STR(TotalReactSaturn)& ” Milliseconds.”)

ENDPROCEDURE

// Task 3

PROCEDURE Records()

DECLARE House AS CHARACTER

DECLARE Age AS INTEGER

DECLARE Counter AS INTEGER

DECLARE CounterRe AS INTEGER

DECLARE Reaction[0:649] AS FLOAT

DECLARE ArrayCounter AS INTEGER

DECLARE ReactMax AS FLOAT

DECLARE ReactAve AS FLOAT

DECLARE Stop AS BOOLEAN

DECLARE Temporart AS INTEGER

House ← ‘’

Age ← 0

Counter ← -0

CounterRe ← -0

ArrayCounter ← 0

ReactMax ← 0.0

ReactAve ← 0.0

Stop ← FALSE

Temporart ← 0

FOR ArrayCounter is 0 to 649

Reaction[ArrayCounter] <- 0.0

ENDFOR

WHILE Stop = False

OUTPUT(“Please enter an age group from 12 to 16”)

Age ← CheckingAge(INT(INPUT))

OUTPUT(“Please enter a house M for Mars and S for Saturn”)

House <- CheckingHouse(UPPERCASE(CHAR(INPUT)))

WHILE SchoolHouse[Counter] =! ‘’

IF Schoolhouse[Counter] = House AND AgeOfStudent[Counter] = Age

Reaction[CounterRe] ← ReactionTime[Counter]

CounterRe = CounterRe + 1

ENDIF

Counter = Counter + 1

ENDWHILE

FOR i 0 TO CounterRe

ReactAve = ReactAve + Reaction[i]

ENDFOR

ReactAve = ReactAve / CounterRe

//Bubble Sort Algorithm

//SWAP is taking both of the values and switching their place with

//eachother

FOR i TO CounterRe

FOR j TO CounterRe - 1

IF Reaction[i] > Reaction[i+1]

SWAP(Reaction[i], Reaction[i+1])

ENDIF

ENDFOR

ENDFOR

ReactMax ← Reaction[CounterRe]

//Output Message

OUTPUT(“The Slowest and the Average Reaction Times for House “

&STR(House)&” at the age group of “&STR(Age)&” Is “&STR(ReactAve)&”

Which is the Average time and “&STR(ReactMax)&” Which is the slowest

time”)

OUTPUT(“Would you like to check another House and Age? 0 for YES,

And 1 for NO “)

Temporart <- INT(INPUT)

WHILE Temporart =! 1 AND Temporart =! 0

OUTPUT(“Please Enter a valid number either 1 For No or 0 For Yes”)

Temporart <- INT(INPUT)

ENDWHILE

Stop ← BOOL(Temporart)

ENDWHILE

ENDPROCEDURE

FUNCTION CheckingAge(Age As INTEGER) RETURN INTEGER

WHILE Age <= 12 OR Age >= 16

IF Age < 11 THEN

OUTPUT(“Age is too small make sure it is more or equal to 12”)

ELSE

OUTPUT(“Age is too big make sure it is smaller or equal to 16”)

ENDIF

Age <- INT(INPUT)

ENDWHILE

RETURN Age

ENDFUNCTION

FUNCTION CheckingHouse(House As CHARACTER) RETURN CHARACTER

WHILE HOUSE =! ‘M’ AND HOUSE =! ‘S’

OUTPUT(“Invalid House Please Enter M for Mars or S for Saturn”)

House <- UPPERCASE(CHAR(INPUT))

ENDWHILE

RETURN House

ENDFUNCTION

FUNCTION CheckingReactionTime(Time As FLOAT) RETURN FLOAT

WHILE Time < 1.0 OR Time > 3000.0

IF Time < 1.0 THEN

OUTPUT(“Reaction Time too small make sure the reaction time is

More than 1.0 milliseconds”)

ELSE

OUTPUT(“Reaction Time too big make sure the reaction time is

Less than 3000.0 milliseconds”)

ENDIF

Time <- FLOAT(INPUT)

ENDWHILE

RETURN Time

ENDFUNCTION

**//Menu is NOT needed in the assignment.**