**Tutorial 7: Functions**

1.

Hierarchy view:

Student Kiosk

Student

Finance

Admin

Enquiry

Institution

Registration

Residence Registration

Pseudocode:

2.

int evenNumbers(){

for (int i=2; i < 100; i++){

if (i%2 = 0){

print(i)

}

}

}

3.

int convertToPounds(nadDollars){

prompt user for exchangeRate and get exchangeRate

pounds = nadDollars / exchangeRate

}

4.

Start:

prompt user for size and get size

int[size] array = []

int max(array[]){

max = array[0]

for (int i=1; i < array.length; i++){

if (max[i] > max[0]){

max = array[i]

}

}

return max

}

int min(array[]){

min = array[0]

for (int i=1; i < array.length; i++){

if( min[i] < min[0]){

min = array[i]

}

}

return min

}

Stop

7. Key points:

* Hiring 46 new recruits
* 16 recruits are female rest are male(30)
* Age of candidates should be between(exclusive by default) 18 and 25
* Women should run 10km in less 60min and men in 45min.

Start

get maxFemaleRecruits = 16

get maxMaleRecruits = 30

get distanceRequired = 10

get femaleFinishTimeRequired = 60

get maleFinishTimeRequired = 45

get femaleCounter = 0

get maleCounter = 0

while(true){

//Check number of recruits

if (femaleCounter+maleCounter == 46){

display “Recruitment space full!”

exit

}

prompt user for candidateGender and get candidateGender

prompt user for candidateAge and get candidateAge

prompt user for candidateCitizenship and get candidateCitizenship

prompt user for distanceCoveredByCandidate and get distanceCoveredByCandidate

prompt user for timeFinished and get timeFinished

case of (candidateGender){

“Female”:

if (femaleCounter > maxFemaleRecruits){

display “No more space for female recruits”

continue

}

if ( 18 < candidateAge < 25) {

if (candidateCitizenship == “Namibian”){

if (distanceCoveredByCandidate >= distanceRequired){

if (timeFinished < femaleFinishedTimeRequired){

femaleCounter++

display “Candidate recruited!”

continue

}

else{

display “Candidate not eligible!”

}

}

else{

display “Candidate not eligible!”

}

}

else{

display “Candidate not eligible”

}

}

else{

display “Candidate not eligible”

}

“Male”:

if (maleCounter > maxMaleRecruits){

display “No more space for male recruits”

continue

}

if ( 18 < candidateAge < 25) {

if (candidateCitizenship == “Namibian”){

if (distanceCoveredByCandidate >= distanceRequired){

if (timeFinished < maleFinishedTimeRequired){

maleCounter++

display “Candidate recruited!”

continue

}

else{

display “Candidate not eligible!”

}

}

else{

display “Candidate not eligible!”

}

}

else{

display “Candidate not eligible”

}

}

else{

display “Candidate not eligible”

}

default:

display “Candidate gender not valid”

endcase

}

Stop

Useful functions:

//Find an item in array

Start

find(array[], arraySize, itemToBeFound){

for(int i=arraySize-1; i > -1; i--){

if(itemToBeFound == array[i]){

display ‘Item found’

return

}

}

display ‘Item:’ + itemToBeFound + ‘ Not found’

}

//How many times does an item appear in array

frequency(array[], arraySize, itemToBeFound){

found = 0

for (int i= arraySize -1; i > -1; i--){

if (itemToBeFound == array[i]){

found++

}

}

display ‘Item found: ’ + found + ‘ times’

}

//Average of numeric values in array

average(array[], arraySize){

sum = 0

for (int i=arraySize -1; i > 0; i--){

sum += array[i]

}

average = sum / arraySize

}

Stop