Callum Simpson B6030326

```
Algorithm:      First Fit Rope Cutter
Inputs:
                orders : Array list of (customers) Integers ,
                coils : array list of ropes //coils ordered from manufacture
Variables:
                I, j : Integer // Flow control
                currentRopesUsed, ropesRemoved; Integer // ropes currently used and ropes removed
Returns:
                currentRopesUsed; Interger
Begin:
        currentRopesUsed := 0 //No ropes used at the start
        ropesRemoved : = 0    //No ropes removed at the start. Used to help with indexing
        for I := 0 to size(orders) - 1 do // go through all the orders
                for j : = 0 to size(coils) - 1 do // go through all the ropes
                        if ropes [j] length >= the order [I] then // can the rope fulfil the order?
                                ropes [j] = ropes [j] - ordes[j] //cut current rope j by the current order size
                                if currentRopeUsed is <= rope[j] then // if it's a new rope
                                        currentRopeUsed := currentRopeUsed +1 //move forward one rope
                                fi
                                if rope j length <= 5 then //is the ropes size less than 5
                                        remove rope[j] // remove the rope
                                        ropesRemoved := ropesRemoved + 1// add one to the removed pile
                                        currentRopeUsed := currentRopeUsed – 1//move back one rope
                                fi
                        break
                        fi
                od
        od
        return currentRopeUsed + ropesRemoved; // the total ropes used
End
```

Callum Simpson B6030326

```
Algorithm:        Next Fit Rope Cutter
Inputs:
                  orders ; Array list of Integers ,
                  coils ; array list of ropes //coils ordered from manufacture
Variable:
                  I :Intger //Flow Control
                  currentRopesUsed: Integer , //number of ropes used
                  removedRopes: Integer //number of coils removed
                  completeOrderCounter: Integer // how many orders have been used
                  slectedRope : Integer
Returns:
                  currentRopesUsed; Interger
Begin:
        currentRopesUsed := 0 //no ropes used
        completeOrderCounter := 0 //no orders completed
        removedRopes := 0 //no ropes used
        slecetedRope := 0// no ropes selected.
        for I := 0 to size(orders) - 1 do
                while(size(orders) > completeOrderCounter) do //while there still orders to complete
                        if (slecetedRope == 0) do // if it's the first rope
                        currentRopesUsed := currentRopesUsed + 1 //increase the current ropes
                        fi
                        if ropes[slecetedRope] length >= the order I then // can the rope fulfil the order?
                                cut rope[slecetedRope] by the current order [i] size //Cut the rope

                                if rope slecetedRope length <= 5 then // is the rope less than 5 meters long
                                        remove rope[slectedRope] // bin the rope
                                        ropesRemoved := ropesRemoved +1 // a rope has been binned
                                fi
                                completeOrderCounter := completeOrderCounter + 1;// move onto the next
                        order
                                break
                        fi
                        else do
                                slectedRope := slectedRope + 1 //move to the next rope ropes by
                                completeOrderCounter := slectedRope +1 //move to the next order
                        od
                od
        fi
        return currentRopesUsed + removedRopes //returns ropes used
od
End
```