Pseudocode FOR Task1, 2 & 3

Task 1

Allow auction company to enter item details.

```
product_selling <-- 0
 OUTPUT("Welcome to auction software.")

REPEAT
    INPUT product_selling
    OUTPUT "Error, please enter more than 10 products."
UNTIL product_selling < 10

name_list <-- []
bid_number_list <-- [0]*product_selling
description_list <-- []
reserve_price_list <-- []
item_number_list <-- list(range(1, product_selling + 1))

FOR counter <-- 1 to length(product_selling)
    INPUT _name, _description, _price
    name_list[counter] <-- _name
    description_list[counter] <-- _description
    reserve_price_list[counter] <-- _price
NEXT
```

Task 2

Allow buyers to purchase

```
highest_bid_list <-- [0.0] * product_selling
buyer_number_list <-- ["1", "2", "3", "4", "5", "6"]
item_highest_bid_holder_list <-- [""]*10
buyer_number_check <-- "0"
cookie <-- False
WHILE purchase_status <-- "no" DO
    OUTPUT "\nWelcome to auction! "
ENDWHILE
    FOR counter_2 <-- 1 TO length(name_list)
        _item_num <-- item_number_list[counter_2]
        current_item_name <-- name_list[counter_2]
        OUTPUT _item_num, ": ", current_item_name
    NEXT
    CASE buyer_number_check OF
        "0": OUTPUT "IF you want to bid, please enter your buyer number" INPUT
buyer_number_check
```

```
        "exit": BREAK
    ENDCASE
    WHILE buyer_number_check IN buyer_number_list DO
        IF NOT cookie
            THEN OUTPUT "Identity verified."
            INPUT item_to_buy
        ENDIF
        IF item_to_buy NOT IN name_list
            THEN OUTPUT "Item number invalid, try again."
            CONTINUE
        ENDIF
        search_index <-- name_list.index(item_to_buy)
        current_description <-- description_list[search_index]
        item_highest_bid <-- highest_bid_list[search_index]
        OUTPUT "Details: ", current_description
        OUTPUT "Current highest bit is ", item_highest_bid

        REPEAT
            INPUT buyer bid
            IF buyer_bid > item_highest_bid
                THEN item_highest_bid <-- buyer_bid
                highest_bid_list[search_index] <-- item_highest_bid
                bid_number_list[search_index] +<-- 1
                item_highest_bid_holder_list[search_index] <-- buyer_number_check
                OUTPUT "Congratulation! Your bid is the current highest bid."
                cookie <-- True
                BREAK
            ELSE
                 OUTPUT("Sorry, bid lower than current highest bid. Try again.")
            ENDIF
            INPUT purchase_status("Do you want to bid FOR another item? Y/N")
        UNTIL purchase_status = "n"
        IF buyer_number_check not IN buyer_number_list
            THEN OUTPUT "Buyer number invalid, try again. "
            buyer_number_check <-- "0"
        ENDIF
ENDWHILE
```

Task 3

Calculate and show statistics

```
highest_price_list <-- []
under_reserve_price_list <-- []
no_bid_list <-- []
sold_status_list <-- ["no"] * product_selling
total_price <-- 0
FOR counter_3 <-- 1 TO length(highest_bid_list)
    IF highest_bid_list[counter_3] = 0
        THEN no_bid_list[counter] <-- counter_3 + 1
        sold_status_list[counter_3] <-- "no"
    ENDIF
```

```
    IF highest_bid_list[counter_3] < reserve_price_list[counter_3] AND
highest_bid_list[counter_3] <> 0
        THEN under_reserve_price_list[counter_3] <-- counter_3 + 1
        sold_status_list[counter_3] <-- "no"
    ENDIF
    IF highest_bid_list[counter_3]) > reserve_price_list[counter_3])
        THEN highest_price_list[counter_3] <-- counter_3 + 1
        sold_status_list[counter_3] <-- "yes"
        total_price <-- total_price + highest_bid_list[counter_3] * 1.1
    ENDIF
NEXT

OUTPUT "Total price is $", total_price
OUTPUT "following item has at least 1 bid, but the bid is lower than the reserve
price:", under_reserve_price_list
OUTPUT "following item has no bid at all: ", no_bid_list

sold_item_quantity <-- length(highest_price_list)
under_reserve_price_item_quantity <-- length(under_reserve_price_list)
no_bid_quantity <-- length(no_bid_list)
OUTPUT sold_item_quantity, " is/are sold."
OUTPUT under_reserve_price_item_quantity, " is/are lower than reserve price."
OUTPUT no_bid_quantity, " has/have no bids."
```