Take-Home Exam 1

1)

To solve this question first of all, I made a use of abstraction.
As we are given a block of information in a paragraph, I concentrated on the essential points from a big data by removing the unnecessary details.
To illustrate, below I will present the information I found important in solving this question:

Overall length of the distance = 3.91 m
1 battery-life lasts for 17 cm distance
Maximum number of batteries 1 tank can hold= 15 batteries
Overall number of batteries that can be utilized= 45 batteries

After obtaining the necessary elements, I made use of decomposition. In question we are given distance in various units. For example, to know exact unit of distance I converted meters to centimetres => (3.91 m= 391cm). Also, I divided the overall distance into smaller parts according to the maximum distance one battery can take so that I can define spots to leave my batteries if needed (as single battery can pass a way of 17 cm, I divided the way into 17 => 391 cm / 17cm= 23). Thus, I am left with much more simplified data in my outline. Now I just have 45 batteries and 23 spots to reach the destination, which is much practical than calculating the length and centimetres.

Thus, having this data I referred another concept of computational thinking, Algorithm design, which is step-by-step set of instructions

1. D-bots takes 15 batteries (remaining batteries in the starting point is 45-15=30);
2. D-bot comes to 3rd spot and leaves 9 batteries on 3rd spot;
3. It goes back with 3 batteries on it. Thus, it comes back to starting point with empty tank;
4. It takes 15 more batteries from the starting point (remaining batteries in the starting point is 30-15 = 15);
5. It goes back to the 3rd spot again and leaves 9 batteries on the 3rd spot (total batteries on the 3rd spot is 9+9 = 18 batteries);
6. it goes back to starting point and takes remaining 15 batteries (remaining batteries on the starting point 15-15= 0);
7. it comes to 3rd spot with 12 batteries, which makes total number of batteries on the 3rd spot = 30, (18+12 = 30);
8. It travels 5 more spots and leaves its 5 batteries on spot #8;
9. It returns back to 3rd spot with empty tank;
10. D-bot takes remaining 15 batteries from the 3rd spot and reaches the 8th spot with 10 batteries;
11. It collects 5 batteries from the spot #8, now D-bot has 15 batteries, which is exactly what it needs to reach the final destination.
12. D-bot reaches the final destination.

While creating the above algorithm, the main motivation was to have minimum of 15 batteries on spot #8, because from spot #8 D-bot needs exactly 15 batteries to reach the end. Now using the decomposition, we break the problem into the sub-problems, which are:

- reaching the point 8 with using-up maximum 30 batteries,
- having minimum of 15 batteries on point 8 to have a full tank,
- as we cannot bring 15 batteries to point 8 directly, we will have to refill the tank of the D-bot twice on some spot before point #8.

My motivation was spending D-bot's batteries on transportation as less as possible and to store more number of batteries on chosen spots. Keeping this in mind, I looked for the spot furthest from point# 8, where from D-bot can come to point 8, leave some batteries and return back to refill its tank, and when it comes to point #8 again to have full tank. Later on, I found out that 5-spot-distance is the longest distance D-bot can go, leave 5 batteries at most, and return back to refill its tank and to have full tank on the point #8 (5 stored batteries + 10 batteries on tank), which is point #3. Although, the 7-spot distance(spot#1) and 6-spot distance(spot#2) were longer, if the D-bot stores its extra batteries there, when coming to point#8, D-Bot can have maximum of 14 batteries which is not enough to reach the end point. A new subproblem is to bring all of the batteries to spot#3 from the starting point, which could be done in these steps: taking 15 batteries, coming to point #3 with 12 batteries , leaving 9 batteries, coming back, taking another 15 batteries, coming to point #3 with 12 batteries , leaving 9 batteries, coming back, getting last 15 batteries from the start position and come to the spot #3 where there will be 30 batteries in total, which will enable the D-bot refill its tank twice, which will enable the d-bot to have full tank on the spot#8 .

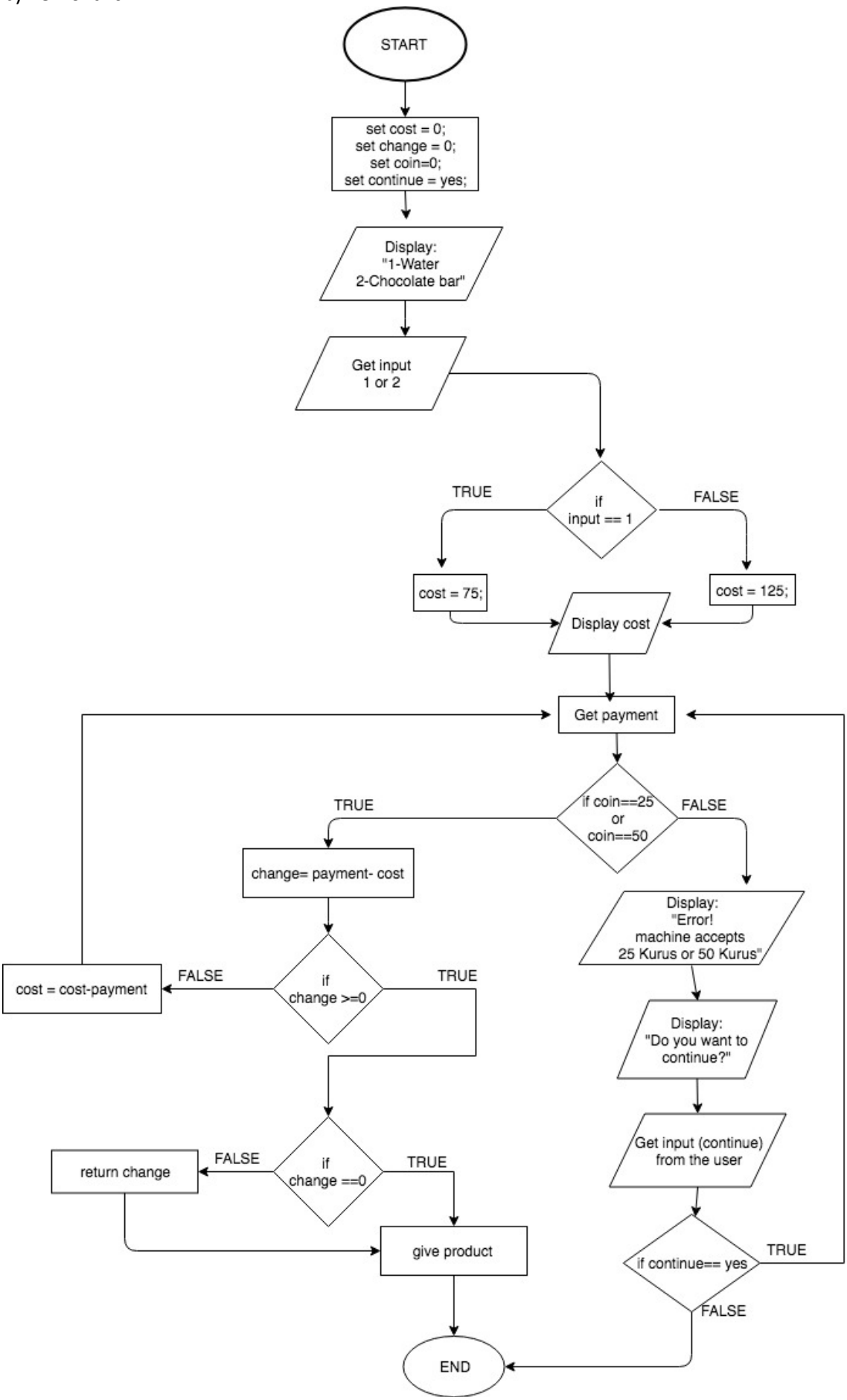
2)

a) **The pseudocode of the algorithm for the vending machine.**

1. Start the program
2. Set cost=0;
3. Set change=0;
4. Set coin = 0;
5. Display input values on the screen: “1-Water, 2-Chocolate bar”
6. Get input from the user
7. If consumer chooses ‘1’
   Assign cost=75
   else
   Assign cost=125
8. Display the cost of the product the user has chosen
9. Getting payment from the user
10. If vending machine gets 25 kurus or 50 kurus
    Calculate: change = payment -cost
    While(change < 0)
    Calculate: cost= cost – payment
    Go to step 9 (Getting payment from the user)
    If change >=0
    If change ==0
    Give the product
    End the program
    else
    Return change
    End the program

    Else
    Display: “Error! Machine accepts only 25 or 50 kurus”
    Ask if consumer wants to continue
    If continue == yes
    Go to step 9 (Getting payment from the user)
    else
    End the program

b)**Flowchart**


START
set cost = 0;
set change = 0;
set coin=0;
set continue = yes;
Display:
"1-Water
2-Chocolate bar"
Get input
1 or 2
if
input == 1
TRUE
FALSE
cost = 75;
cost = 125;
Display cost
Get payment
if coin==25
or
coin==50
TRUE
FALSE
change= payment- cost
Display:
"Error!
machine accepts
25 Kurus or 50 Kurus"
if
change >=0
FALSE
TRUE
cost = cost-payment
Display:
"Do you want to
continue?"
if
change ==0
FALSE
TRUE
return change
Get input (continue)
from the user
give product
if continue== yes
TRUE
FALSE
END