INTRODUCTION

In this task, we have to make a vending machine with some choice of drinks with prices. Once a drink is used above some threshold amount, it has to ask the user for some other drink. And once all drinks have been used to the threshold value, we have to refill the vending machine. Only thing here is one has to implement the above using FINITE-STATE-MACHINES and not the clumsy if-else statement.

PROBLEM STATEMENT

(i) The vending machine will show a list of juices along with their price and code.

(ii)User will enter his choice of drink using the 4 digit code.

(iii)The user will enter the amount of money fed to the vending machine, and will get the change as per the transaction. If he enters a price less than the cost price, some suitable warning has to be displayed.

(iv)Initialize each variety of juice to be 50.Once all the 50 cans are exhausted, it has to display some suitable warning and ask for some new choice.

(v)If all the cans are exhausted, user has to enter REFILL which will replenish all the cans to their original amount, and the vending machine can be used again.

Serial No.	drink	code	cost
1	pepsi	PEPS	30
2	Mountain dew	MDEW	30
3	Dr. pepper	DPEP	50
4	coke	COKE	20
5	gatorade	GATO	20
6	Diet coke	DCOK	30
7	Minute maid	MINM	25
8	tropicana	TROP	30

ALGORITHM USED

This problem was only about the implementation of finite state machines. So I first created the init function which had a dictionary of all the juices with the price, the code and the quantity remaining. As dictionaries are implemented using hash tables, searching keys in them takes O(1) complexity. Next for the transitions are defined using the functions your_choice, money_process, cans_process, refill_juices and run. The current_state transitions to each of the functions as per they are called. The only conditionals used are if the amount entered is less than the juice amount or not, and if the cans have reached threshold value, ie, 50. I have added one extra if statement for if the user enters some wrong code, then a warning is generated to enter another one.

DIFFICULTIES

Learning the finite state machine in itself was a difficult task, and after that thinking of the transitions states to be employed here especially to refill and decrement the quantities. The only thing I couldn't sort out till now is how to completely rule out the if statement, as I had to use it 2 times(my case, 3).

RESULTS AND OBSERVATION

To check the correctness of the code, I changed the quantity to 2, and the code worked correctly.It asked for choices again and again until a particular can was finished. When it is finished, it asks again for the choices, and once all are used 2 items unless you type refill it will keep asking you to enter a choice.

CONCLUSION

This has a future in ARK as many times, we have to understand the object and as per conditions of the object we have to perform required things. Often those if-else gets clumsy, then comes the FINITE-STATE-MACHINES.