MACHAKOS UNIVERSITY

DENNIS KIBET

J17-5580-2020

BSC COMPUTER SCIENCE

AUTOMATA THEORY

SCO211

SCHOOL OF ENGINEERING AND TECHNOLOGY

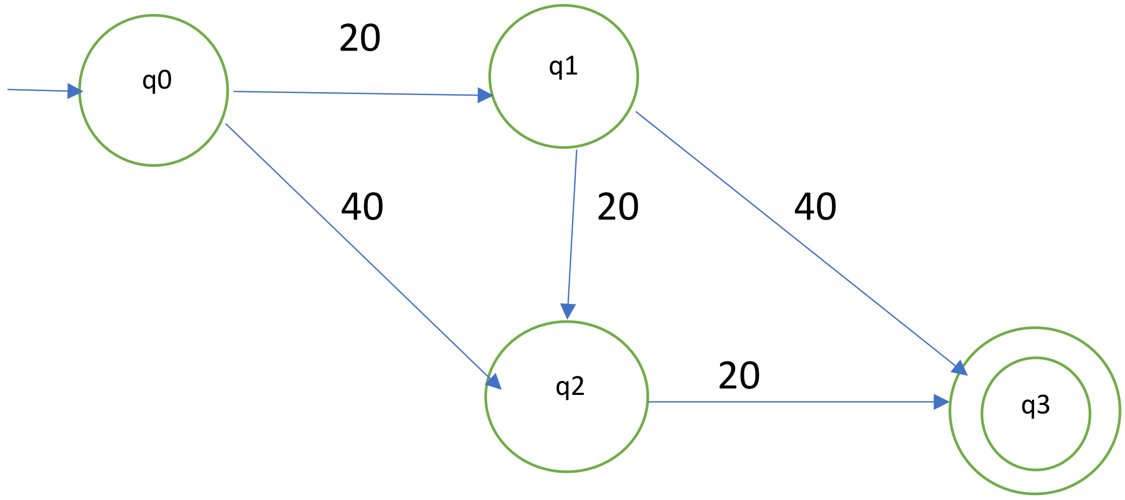
Question

MU cafeteria has installed an ice cream vending machine to automatically dispense ice cream to students and staff. The cost of a can of ice cream is Kshs 60 and the machine only accepts coins in denomination of 20 & 40 only and the machine does not give change.

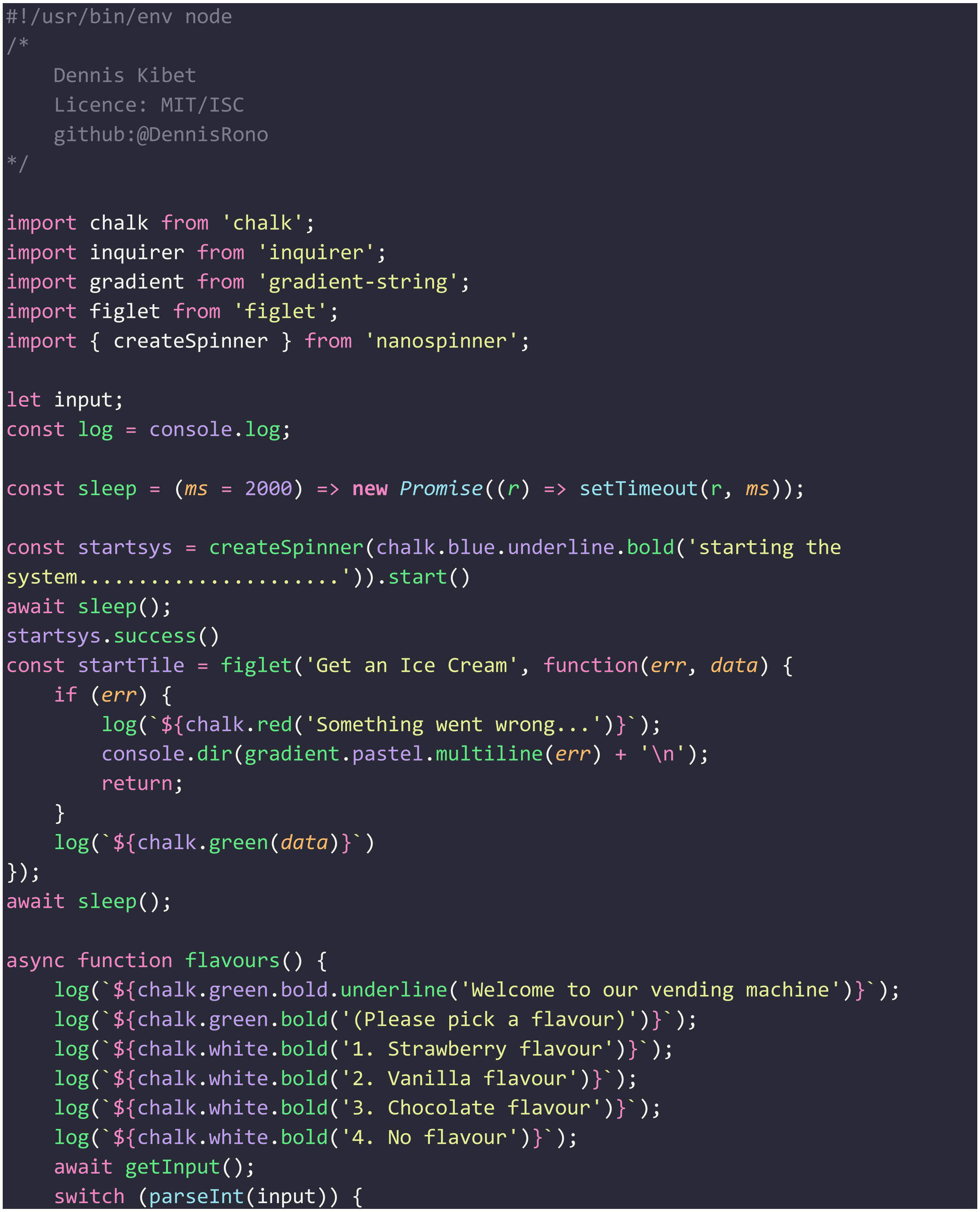
Assignment

a) Formally define this machine as a finite automata . i.e., determine the states, alphabet and the transition function.
   Vending Machine=(Q,∑, δ,q1,F), where
   -Transition functions:
   δ(q0,20)=q1, δ(q0,40)=q2
   δ(q1,20)=q2, δ(q1,40)=q3
   δ(q2,20)=q2
   δ(q2,20)=q3

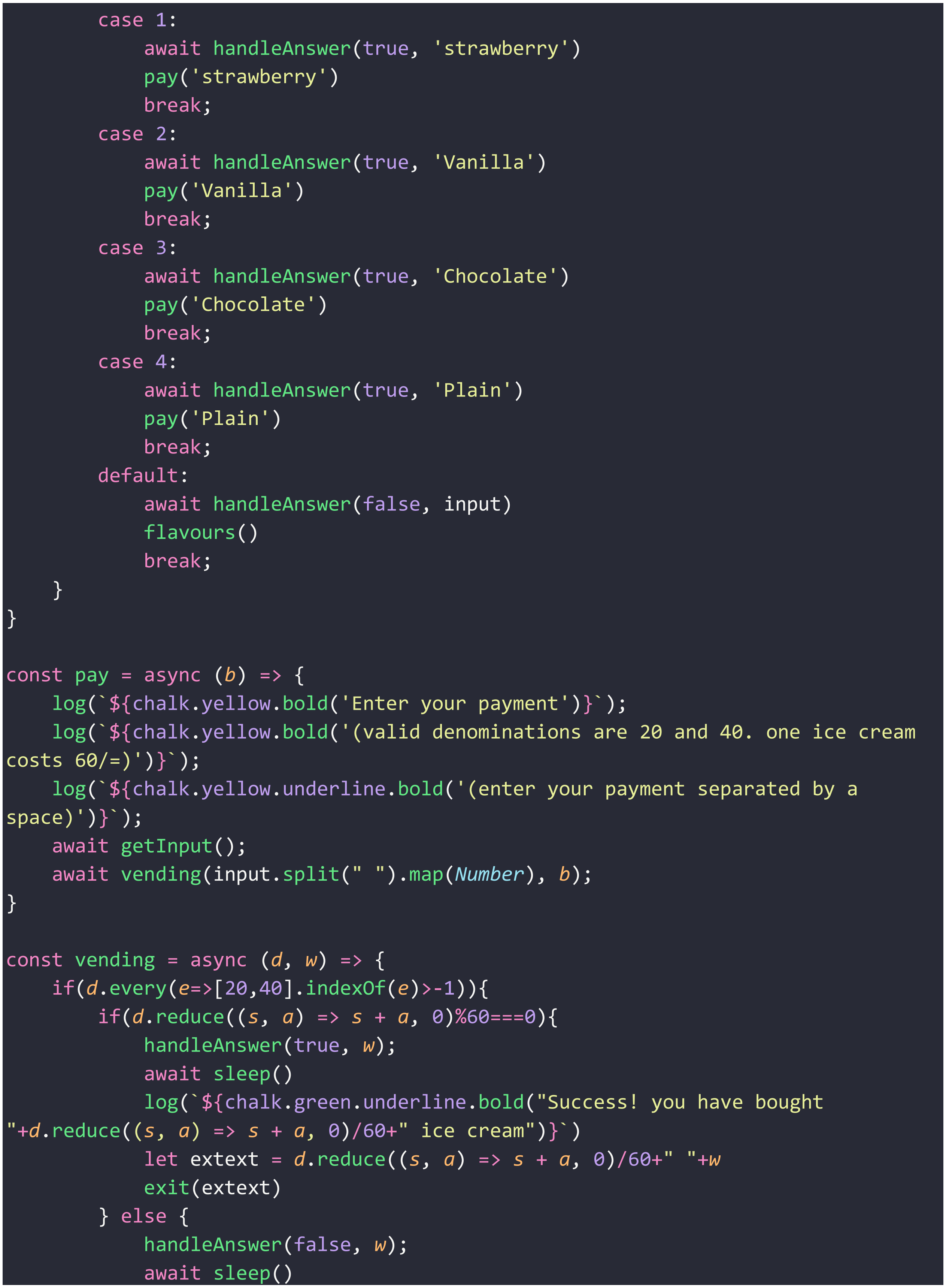
b) Draw the state transition diagram for the machine.



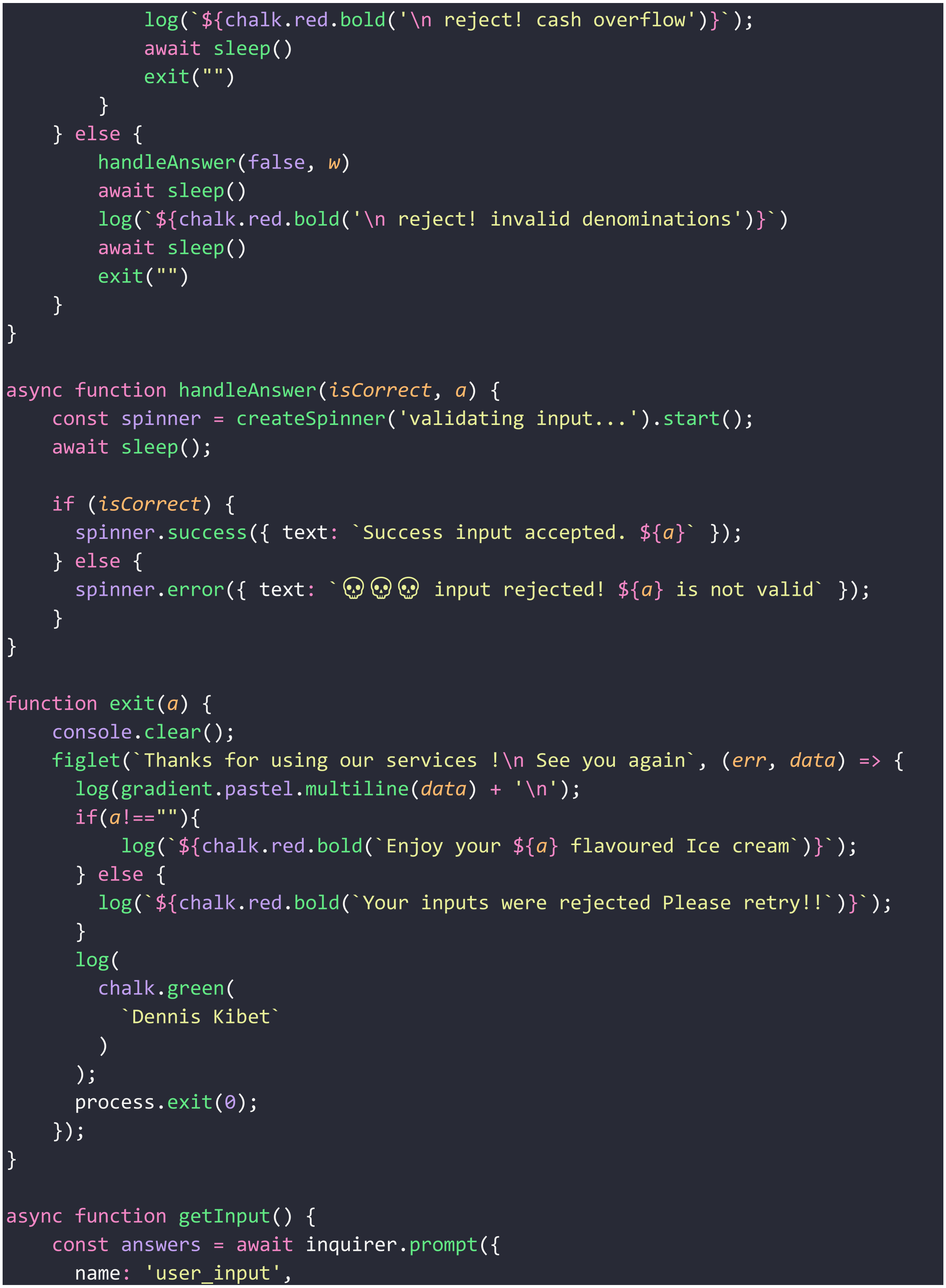
c) Using a programming language of your choice, write a program to simulate the operation of this machine.



```
#!/usr/bin/env node
/*
    Dennis Kibet
    Licence: MIT/ISC
    github:@DennisRono
*/

import chalk from 'chalk';
import inquirer from 'inquirer';
import gradient from 'gradient-string';
import figlet from 'figlet';
import { createSpinner } from 'nanospinner';

let input;
const log = console.log;

const sleep = (ms = 2000) => new Promise((r) => setTimeout(r, ms));

const startsys = createSpinner(chalk.blue.underline.bold('starting the
system......................')).start()
await sleep();
startsys.success()
const startTile = figlet('Get an Ice Cream', function(err, data) {
    if (err) {
        log(`${chalk.red('Something went wrong...')}`);
        console.dir(gradient.pastel.multiline(err) + '\n');
        return;
    }
    log(`${chalk.green(data)}`)
});
await sleep();

async function flavours() {
    log(`${chalk.green.bold.underline('Welcome to our vending machine')}`);
    log(`${chalk.green.bold('(Please pick a flavour)')}`);
    log(`${chalk.white.bold('1. Strawberry flavour')}`);
    log(`${chalk.white.bold('2. Vanilla flavour')}`);
    log(`${chalk.white.bold('3. Chocolate flavour')}`);
    log(`${chalk.white.bold('4. No flavour')}`);
    await getInput();
    switch (parseInt(input)) {
```



```
        case 1:
            await handleAnswer(true, 'strawberry')
            pay('strawberry')
            break;
        case 2:
            await handleAnswer(true, 'Vanilla')
            pay('Vanilla')
            break;
        case 3:
            await handleAnswer(true, 'Chocolate')
            pay('Chocolate')
            break;
        case 4:
            await handleAnswer(true, 'Plain')
            pay('Plain')
            break;
        default:
            await handleAnswer(false, input)
            flavours()
            break;
    }
}

const pay = async (b) => {
    log(`${chalk.yellow.bold('Enter your payment')}`);
    log(`${chalk.yellow.bold('(valid denominations are 20 and 40. one ice cream 
costs 60/=)')}`);
    log(`${chalk.yellow.underline.bold('(enter your payment separated by a 
space)')}`);
    await getInput();
    await vending(input.split(" ").map(Number), b);
}

const vending = async (d, w) => {
    if(d.every(e=>[20,40].indexOf(e)>-1)){
        if(d.reduce((s, a) => s + a, 0)%60===0){
            handleAnswer(true, w);
            await sleep()
            log(`${chalk.green.underline.bold("Success! you have bought 
"+d.reduce((s, a) => s + a, 0)/60+" ice cream")}`)
            let extext = d.reduce((s, a) => s + a, 0)/60+" "+w
            exit(extext)
        } else {
            handleAnswer(false, w);
            await sleep()
```



```
            log(`${chalk.red.bold('\n reject! cash overflow')}`);
            await sleep()
            exit("")
        }
    } else {
        handleAnswer(false, w)
        await sleep()
        log(`${chalk.red.bold('\n reject! invalid denominations')}`)
        await sleep()
        exit("")
    }
}

async function handleAnswer(isCorrect, a) {
    const spinner = createSpinner('validating input...').start();
    await sleep();

    if (isCorrect) {
      spinner.success({ text: `Success input accepted. ${a}` });
    } else {
      spinner.error({ text: `💀💀💀 input rejected! ${a} is not valid` });
    }
}

function exit(a) {
    console.clear();
    figlet(`Thanks for using our services !\n See you again`, (err, data) => {
      log(gradient.pastel.multiline(data) + '\n');
      if(a!==""){
          log(`${chalk.red.bold(`Enjoy your ${a} flavoured Ice cream`)}`);
      } else {
        log(`${chalk.red.bold(`Your inputs were rejected Please retry!!`)}`);
      }
      log(
        chalk.green(
          `Dennis Kibet`
        )
      );
      process.exit(0);
    });
}

async function getInput() {
    const answers = await inquirer.prompt({
      name: 'user_input',
      type: 'input',
      message: `${chalk.red.bold('>>> ')}`,
      default() {
        return '1';
      },
    });

    input = answers.user_input;
  }

console.clear();
flavours();
```