Aganitha Web Development Quiz 2023-24

musafarshakeel@gmail.com Switch account

* Indicates required question

Technical Reading Comprehension - 2

Commander is a popular npm package used for building command-line interfaces (CLIs) in Node.js applications. Let's take a look at a basic example to understand how Commander works.

```
const { Command } = require('commander');
const program = new Command();

program
  .description('An application for pizza ordering')
  .option('-p, --peppers', 'Add peppers')
  .option('-c, --cheese <type>', 'Add the specified type of cheese'
, 'marble')
  .option('-C, --no-cheese', 'You do not want any cheese');

program.parse();

const options = program.opts();
console.log('You ordered a pizza with:');
if (options.peppers) console.log('  - Peppers');
const cheese = !options.cheese ? 'no' : options.cheese;
console.log('  - %s cheese', cheese);
```

In this example, we are building a pizza ordering CLI application using Commander. The program accepts various options to customize the pizza. We have defined three options:

- -p, --peppers: Adds peppers to the pizza.
- -c, --cheese <type>: Adds the specified type of cheese to the pizza. By default, it uses 'marble' cheese if no type is provided.
- -C, --no-cheese: Indicates that you do not want any cheese on your pizza.

After parsing the command-line arguments with program.parse(), the options are stored in the options object. The program then displays the pizza order based on the selected options.

To invoke this program from the command line, you would use the following command:

```
   node pizza.js --peppers --cheese=cheddar
```

The expected output would be:

```
You ordered a pizza with:
  - Peppers
  - cheddar cheese
```

You can define a boolean option long name with a leading no- to set the option value to false when used. Defined alone this also makes the option true by default
----

```
const { Command } = require('commander');
const program = new Command();
program
  .description('An application for pizza ordering')
  .option('-p, --peppers', 'Add peppers')
  .option('-c, --cheese <type>', 'Add the specified type of cheese',
'marble')
  .option('-C, --no-cheese', 'You do not want any cheese');

program.parse();
const options = program.opts();
console.log('You ordered a pizza with:');
if (options.peppers) console.log('  - Peppers');
const cheese = !options.cheese ? 'no' : options.cheese;
console.log('  - %s cheese', cheese);
```

```
command line: pizza-options
expected output: You ordered a pizza with sauce and mozzarella
cheese
command line: pizza-options --cheese=blue
expected output: You ordered a pizza with sauce and blue cheese
command line: pizza-options --no-sauce --no-cheese
expected output: You ordered a pizza with no sauce and no cheese
```

----

1. How would you modify the Commander definition to add a boolean drinks option to the pizza ordering program? Select all that may apply. * 2 points

- [ ] .option('-d, --drinks', 'Add a drink to the order')
- [ ] .option('-d, ---drinks', 'Add a drink to the order')
- [ ] .option('-d, --drinks <type>', 'Add a drink to the order')
- [ ] .option('-d, --drinks-type', 'Add a drink to the order')

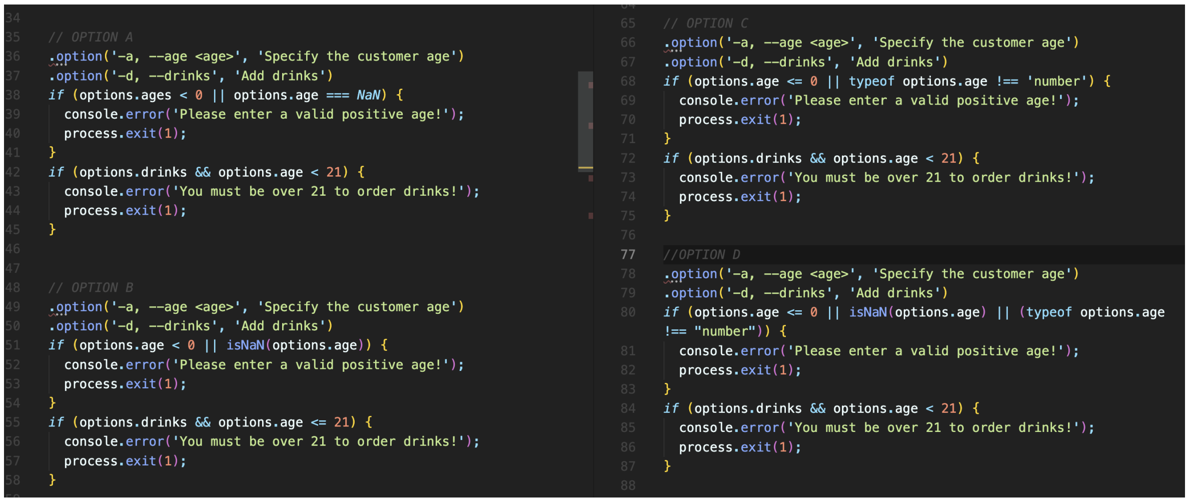
2. How would you modify the Commander definition to add a drinkType option (without a default value) and ensure that it throws an error when the drinks option is provided without specifying the drinkType? Refer to figure 2 below to pick your answer. Select all that may apply. * 2 points

- [ ] Option A
- [ ] Option B
- [ ] Option C
- [ ] Option D

3. How would you modify the Commander definition to add an age option and immediately reject the order with a message if the age is below 21 when a drink is selected (regardless of the drink type)? Also, check whether the age entered is a non zero positive number or not, and throw an error message if it isn't. Refer to figure 3.Select all that apply. 2 points

- ◯ Option A
- ◯ Option B
- ◯ Option C
- ◯ Option D

Figure 3



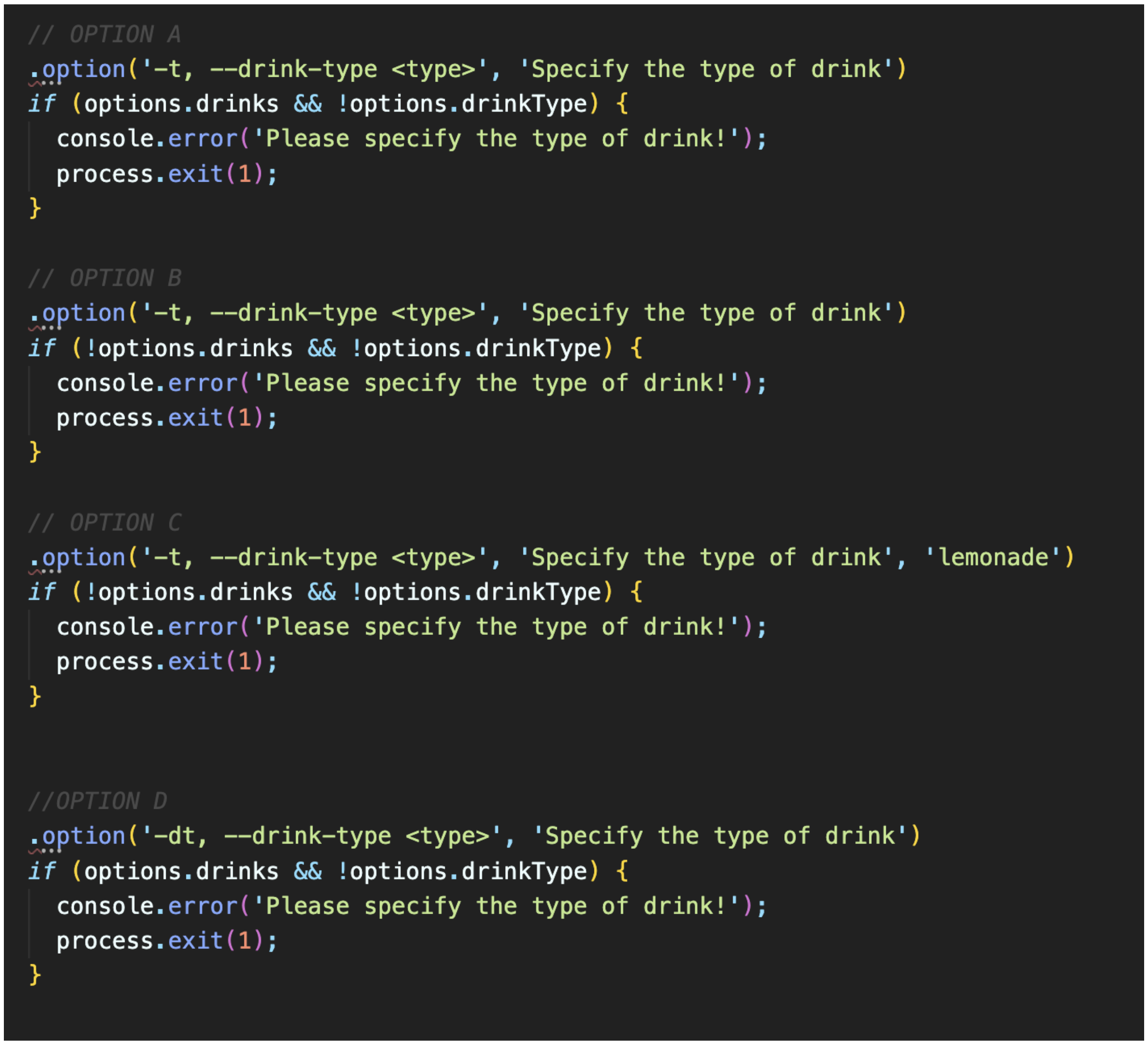
```
// OPTION A
.option('-a, --age <age>', 'Specify the customer age')
.option('-d, --drinks', 'Add drinks')
if (options.ages < 0 || options.age === NaN) {
  console.error('Please enter a valid positive age!');
  process.exit(1);
}
if (options.drinks && options.age < 21) {
  console.error('You must be over 21 to order drinks!');
  process.exit(1);
}


// OPTION B
.option('-a, --age <age>', 'Specify the customer age')
.option('-d, --drinks', 'Add drinks')
if (options.age < 0 || isNaN(options.age)) {
  console.error('Please enter a valid positive age!');
  process.exit(1);
}
if (options.drinks && options.age <= 21) {
  console.error('You must be over 21 to order drinks!');
  process.exit(1);
}
```

```
// OPTION C
.option('-a, --age <age>', 'Specify the customer age')
.option('-d, --drinks', 'Add drinks')
if (options.age <= 0 || typeof options.age !== 'number') {
  console.error('Please enter a valid positive age!');
  process.exit(1);
}
if (options.drinks && options.age < 21) {
  console.error('You must be over 21 to order drinks!');
  process.exit(1);
}

//OPTION D
.option('-a, --age <age>', 'Specify the customer age')
.option('-d, --drinks', 'Add drinks')
if (options.age <= 0 || isNaN(options.age) || (typeof options.age
!== "number")) {
  console.error('Please enter a valid positive age!');
  process.exit(1);
}
if (options.drinks && options.age < 21) {
  console.error('You must be over 21 to order drinks!');
  process.exit(1);
}
```

Figure 2



```
// OPTION A
.option('-t, --drink-type <type>', 'Specify the type of drink')
if (options.drinks && !options.drinkType) {
  console.error('Please specify the type of drink!');
  process.exit(1);
}

// OPTION B
.option('-t, --drink-type <type>', 'Specify the type of drink')
if (!options.drinks && !options.drinkType) {
  console.error('Please specify the type of drink!');
  process.exit(1);
}

// OPTION C
.option('-t, --drink-type <type>', 'Specify the type of drink', 'lemonade')
if (!options.drinks && !options.drinkType) {
  console.error('Please specify the type of drink!');
  process.exit(1);
}


//OPTION D
.option('-dt, --drink-type <type>', 'Specify the type of drink')
if (options.drinks && !options.drinkType) {
  console.error('Please specify the type of drink!');
  process.exit(1);
}
```

Back Next Clear form