1) Create a driver program to create three pizzas one of each size with 3, 6, and 9 toppings to your liking and eat() all of them

```
Pizza smallPizza = new PizzaHutBuilder()
        .setSize(pizzaSize:"small")
        .setPepperoni(hasPepperoni:true)
        .setMushrooms(hasMushrooms:true)
        .setExtraCheese(hasExtraCheese:true)
        .build();
smallPizza.eat();

Pizza mediumPizza = new PizzaHutBuilder()
        .setSize(pizzaSize:"medium")
        .setSausage(hasSausage:true)
        .setOnions(hasOnions:true)
        .setPeppers(hasPeppers:true)
        .setOlives(hasOlives:true)
        .setSpinach(hasSpinach:true)
        .setTomatoAndBasil(hasTomatoAndBasil:true)
        .build();
mediumPizza.eat();

Pizza largePizza = new PizzaHutBuilder()
        .setSize(pizzaSize:"large")
        .setPepperoni(hasPepperoni:true)
        .setBacon(hasBacon:true)
        .setChicken(hasChicken:true)
        .setBeef(hasBeef:true)
        .setHam(hasHam:true)
        .setPesto(hasPesto:true)
        .setSpicyPork(hasSpicyPork:true)
        .setHamAndPineapple(hasHamAndPineapple:true)
        .setExtraCheese(hasExtraCheese:true)
        .build();
largePizza.eat();
System.out.println();
```

Output :

```
Pizza Hut - small Pizza
Toppings:
- Pepperoni
- Mushrooms
- Extra Cheese
Pizza Hut - medium Pizza
Toppings:
- Sausage
- Onions
- Peppers
- Olives
- Spinach
- Tomato and Basil
Pizza Hut - large Pizza
Toppings:
- Pepperoni
- Bacon
- Extra Cheese
- Chicken
- Beef
- Ham
- Pesto
- Spicy Pork
- Ham and Pineapple
```

2) Now assume you purchased another two pizza chains, Little Caesars, and Dominos. You want to add them to your program following the rules mentioned above.

```
Pizza pizza1 = new PizzaHutBuilder()
    .setSize(pizzaSize:"large")
    .setPepperoni(hasPepperoni:true)
    .setSausage(hasSausage:true)
    .setMushrooms(hasMushrooms:true)
    .build();

Pizza pizza2 = new PizzaHutBuilder()
    .setSize(pizzaSize:"small")
    .setPepperoni(hasPepperoni:true)
    .setSausage(hasSausage:true)
    .build();


Pizza pizza3 = new LittleCaesarsBuilder()
    .setSize(pizzaSize:"medium")
    .setPepperoni(hasPepperoni:true)
    .setSausage(hasSausage:true)
    .setMushrooms(hasMushrooms:true)
    .setBacon(hasBacon:true)
    .setOnions(hasOnions:true)
    .setExtraCheese(hasExtraCheese:true)
    .setOlives(hasOlives:true)
    .setBeef(hasBeef:true)
    .build();


Pizza pizza4 = new LittleCaesarsBuilder()
    .setSize(pizzaSize:"small")
    .setPepperoni(hasPepperoni:true)
    .setSausage(hasSausage:true)
    .setMushrooms(hasMushrooms:true)
    .setBacon(hasBacon:true)
    .setOnions(hasOnions:true)
    .setBeef(hasBeef:true)
    .build();

Pizza pizza5 = new DominosBuilder()
    .setSize(pizzaSize:"small")
    .setPepperoni(hasPepperoni:true)
    .build();

Pizza pizza6 = new DominosBuilder()
    .setSize(pizzaSize:"large")
    .setPepperoni(hasPepperoni:true)
    .setSausage(hasSausage:true)
    .setMushrooms(hasMushrooms:true)
    .build();
```

Output:

```
Pizza Hut - large Pizza
Toppings:
- Pepperoni
- Sausage
- Mushrooms
Pizza Hut - small Pizza
Toppings:
- Pepperoni
- Sausage
Little Caesars - medium Pizza
Toppings:
- Pepperoni
- Sausage
- Mushrooms
- Bacon
- Onions
- Extra Cheese
- Olives
- Beef
Little Caesars - small Pizza
Toppings:
- Pepperoni
- Sausage
- Mushrooms
- Bacon
- Onions
- Beef
Dominos - small Pizza
Toppings:
- Pepperoni
Dominos - large Pizza
Toppings:
- Pepperoni
- Sausage
- Mushrooms
```

Part 2

```java
public static void main(String[] args){
    Customer[] customers = {
        new Customer(name:"John", mealType:"NoRestriction"),
        new Customer(name:"Alice", mealType:"Paleo"),
        new Customer(name:"Bob", mealType:"Vegan"),
        new Customer(name:"Emily", mealType:"NutAllergy"),
        new Customer(name:"Mike", mealType:"NoRestriction"),
        new Customer(name:"Sarah", mealType:"Paleo")
    };


    for (Customer customer : customers){
        createMealPlan(customer);
    }


}
```

```java
public static void createMealPlan(Customer customer){
    MacronutrientFactory carbsFactory =  new CarbsFactory();
    MacronutrientFactory proteinFactory =  new ProteinFactory();
    MacronutrientFactory fatsFactory =  new FatFactory();

    Macronutrient carb = carbsFactory.createItem(customer);
    Macronutrient protein = proteinFactory.createItem(customer);
    Macronutrient fats = fatsFactory.createItem(customer);

    System.out.println("Customer : " + customer.getName() + ", Diet Plan: " + customer.getMealType() + "\n");
    System.out.println("Carbs: " + carb.getName() + "\n");
    System.out.println("Protein: " + protein.getName() + "\n");
    System.out.println("Fats: " + fats.getName() + "\n\n");
}
```

Was confused on how to implement abstract factory and factory patterns while using Fat, Protein, and Carb factories. I implemented it another way but it didn't match the assignment instructions so I left it as this.

Output:

```
Customer : John, Diet Plan: NoRestriction
Carbs: Pistachio
Protein: Tofu
Fats: Tuna

Customer : Alice, Diet Plan: Paleo
Carbs: Pistachio
Protein: Fish
Fats: Avocado

Customer : Bob, Diet Plan: Vegan
Carbs: Pistachio
Protein: Tofu
Fats: Peanuts

Customer : Emily, Diet Plan: NutAllergy
Carbs: Bread
Protein: Chicken
Fats: SourCream

Customer : Mike, Diet Plan: NoRestriction
Carbs: Pistachio
Protein: Beef
Fats: Tuna

Customer : Sarah, Diet Plan: Paleo
Carbs: Pistachio
Protein: Chicken
Fats: Peanuts
```