


Assignment Case	
DS using CH3	
Periode Berlaku Semester Genap 2023/2024 Valid on Even odd Year 2023/2024	Software Laboratory Center Assistant Recruitment 24-2

Note: Please focus on the main logic and main feature!
(Splash screen and design are not scored)

Soal

Case

BerISHka

Berishka epitomizes the essence of contemporary fashion, blending modern aesthetics with timeless elegance. Our meticulously curated collection showcases an eclectic mix of apparel, ranging from casual chic to sophisticated glamour, designed to inspire and empower individuals to embrace their personal style with confidence.

Driven by a commitment to quality and innovation, Berishka offers a seamless shopping experience where every garment tells a story of craftsmanship and refinement. With a focus on customer satisfaction and attention to detail, we invite you to explore our diverse range of fashion-forward pieces and elevate your wardrobe with Berishka's signature style.

➤ Home Page

- This menu contains of 5 menus, which are **Add, View, Update, Remove** and **Exit**.
- Prompt** user to **input chosen menu**. **Validate** the input must **between 1 and 4 inclusively**.

```

/$$$$$$
| $ _ $ $
| $ $ \ $ $ /$$$$$ /$$$$$ | $ $ / $ _ $ $ | $ $ / $ $ / $$$$$$
| $$$$$$ / $ _ $ $ / $ _ $ $ | $ $ | $ $ $ $ $ | $ $ _ $ $ | $ $ / $ $ /
| $ _ $ $ | $$$$$$ | $ $ \ _ / $ $ \ _ $ $ | $ $ \ $ $ | $$$$$$ / $$$$$$
| $ $ \ $ $ | $ _ _ _ / $ $ | $ $ / $ $ \ $ $ | $ $ | $ $ | $ _ $ $ / $ _ $ $
| $$$$$$ / | $$$$$$ | $ $ / $$$$$$ | $$$$$$ / $ $ | $ $ | $ $ \ $ $ | $$$$$$
| _ _ _ / \ _ _ _ / | _ _ _ / \ _ _ _ / | _ _ / | _ _ / \ _ _ / \ _ _ _ /
1. Add Product
2. View Product
3. Update Product
4. Remove Product
0. Exit
>>

```

Figure 1. Home Page

1. If user choose **Add Product (Menu 1)**, then:

- **Prompt** user to **input product name**. **Validate** the input must be **10-20 characters** long **(inclusive)**.
- **Prompt** user to **input product price**. **Validate** the input must be **between 50000 – 100000 (inclusive)**.
- **Prompt** user to **input product color**. **Validate** the input must be either **Black/White/Brown (case sensitive)**.
- **Prompt** user to **input product quantity**. **Validate** the input must be **between 10 – 99 (inclusive)**.
- **Prompt** user to **input product size**. **Validate** the input must be either **S/M/L/XL (case sensitive)**.

```
Input product name [10 - 20 characters]: Veka Jean
Input product name [10 - 20 characters]: Veka Jeans
Input product price [50000 - 100000]: 49999
Input product price [50000 - 100000]: 101000
Input product price [50000 - 100000]: 60000
Input product color [Black | White | Brown]: black
Input product color [Black | White | Brown]: White
Input product quantity [10 - 99]: 9
Input product quantity [10 - 99]: 101
Input product quantity [10 - 99]: 15
Input product size [S | M | L | XL]: xL
Input product size [S | M | L | XL]: XL
```

Figure 2. Add Product

- After validating all the product's information, **generate the Product ID** based on this format:

BRXXX

X = Random number (0-9)

- Then, insert the new product into file **"products.txt"** with the following format.

ID#ProductName#ProductPrice#ProductColor#Quantity#Size

2. If user choose **View Product (Menu 2)**, then:

- **Read** data from **"products.txt"** with the following format.

ID#ProductName#ProductPrice#ProductColor#Quantity#Size

- **Display** all products from “**products.txt**” and the sorting options.

ID	Product Name	Color	Size	Quantity	Price
BR543	Aanor Jeans	Black	L	10	75000
BR435	Ronron Tshirt	White	M	70	88888
BR284	Veka Jeans	White	XL	15	60000
BR259	Eren Jacket	Black	XL	99	99999

Sort by:
 [1] ID
 [2] Name
 [3] Color
 [4] Quantity
 [5] Size
 [0] Back
 >>

Figure 3. Add Product

- If there is **no data to be displayed**, please **show empty message** to the user.

```

=====
|               There's no product yet!               |
=====
Press [ENTER] to continue...

```

Figure 4. Empty Message

- **Prompt** user to input based on the options. Make sure the products list will be **sorted ascendingly** with the following conditions:
 - **[1]** -> sort based on the **ID**
 - **[2]** -> sort based on the **Product Name**
 - **[3]** -> sort based on the **Product Color**
 - **[4]** -> sort based on the **Product Quantity**
 - **[5]** -> sort based on the **Product Size**
- **Display** the sorted result after every sorting process is completed.

```
Sorted by : ID
=====
| ID | Product Name | Color | Size | Quantity | Price |
=====
| BR259 | Eren Jacket | Black | XL | 99 | 99999 |
| BR284 | Veka Jeans | White | XL | 15 | 60000 |
| BR435 | Ronron Tshirt | White | M | 70 | 88888 |
| BR543 | Aanor Jeans | Black | L | 10 | 75000 |
=====

Sort by:
[1] ID
[2] Name
[3] Color
[4] Quantity
[5] Size
[0] Back
>>
```

Figure 5. Sorted Result

3. If user choose **Update Product (Menu 3)**, then:

- **Display** all products from “products.txt”
- If there is **no data to be displayed**, please **show empty message** to the user.

```
=====
|           There's no product yet!           |
=====

Press [ENTER] to continue...
```

Figure 6. Empty Message

- **Prompt** user to input **product ID** to update.

```
=====
| ID | Product Name | Color | Size | Quantity | Price |
=====
| BR259 | Eren Jacket | Black | XL | 99 | 99999 |
| BR284 | Veka Jeans | White | XL | 15 | 60000 |
| BR435 | Ronron Tshirt | White | M | 70 | 88888 |
| BR543 | Aanor Jeans | Black | L | 10 | 75000 |
=====

Input product ID to update:
```

Figure 7. Update Product

- If product ID is not found, **display** message to user.

```
Input product ID to update: BR123
Product with ID BR123 not found!

Press [ENTER] to continue...
|
```

Figure 8. Error Message

- If product ID is **found**, **prompt** the user to insert updated product details. **Make sure** to validate all the details **as the same as** the input validation.

```
Enter updated product name [10 - 20 characters]: Enel Jeans
Enter updated product price [50000 - 100000]: 75000
Enter updated product color [Black | White | Brown]: Black
Enter updated product quantity [10 - 99]: 11
Enter updated product size [S | M | L | XL]: L
Product with ID BR284 updated successfully!

Press [ENTER] to continue...
```

Figure 9. Update Product

- Finally, **update** the product details in the file.
4. If user choose **Remove Product (Menu 4)**, then:
- **Display** all products from **“products.txt”**
 - If there is **no data to be displayed**, please **show empty message** to the user.

```
=====
|               There's no product yet!               |
=====

Press [ENTER] to continue...
```

Figure 10. Empty Message

- **Prompt** user to input product ID.

```
=====
| ID | Product Name | Color | Size | Quantity | Price |
=====
| BR259 | Eren Jacket | Black | XL | 99 | 99999 |
| BR284 | Enel Jeans | Black | L | 11 | 75000 |
| BR435 | Ronron Tshirt | White | M | 70 | 88888 |
| BR543 | Aanor Jeans | Black | L | 10 | 75000 |
=====

Input product ID to remove:
```

Figure 11. Remove Product

- If product ID is not found in the list, **display** error message to user.

```
Input product ID to remove: BR123
Product not found!

Press [ENTER] to continue...
```

Figure 12. Error Message

- If product ID **is found**, delete the product in “**product.txt**”, and **display success message**.

```
Input product ID to remove: BR435
Product successfully deleted!
Press [ENTER] to continue...
```

Figure 13. Success Message

5. If user choose **Exit**, then stop the program.

Please run the EXE file to see the sample program.