CR4 – Combinable Coupons with MAX Discount

The GoodPrice store's marketing team never rests!
To increase customer loyalty, they are introducing new coupons: **MAX coupons**, which can be combined with each other, but only up to a specified maximum discount!

These coupons primarily apply to apples, but in the future, similar types are expected for other products as well.
The system must now be capable of processing these.

New Coupon Types

- **A5-MAX10**
  - 5% discount on the total apple price
  - Combinable with other A* or A*-MAX* coupons
  - Maximum discount on apples: 10%
- **A5-MAX15**
  - 5% discount on the total apple price
  - Combinable with other A* or A*-MAX* coupons
  - Maximum discount on apples: 15%

If multiple MAX coupons appear, the system considers the strictest MAX percentage.
For example: if there is one A5-MAX10 and one A5-MAX15, then at most 10% discount can be given in total.
If there are several coupons, at least one MAX coupon is required for combination; however, it does not matter where in the order the MAX coupon is placed.

Usage Rules

- Combination is allowed, but only between coupons for the same product type (e.g., apples).
- The system processes coupons in the given order.
- The discounts add up (e.g., A5 + A5-MAX15 = 10%), but cannot exceed the strictest MAX value.
- If another promotion alone provides a better discount than the coupons combined, the coupons are not used.
- The system returns unused coupons.
- To validate combination, at least one MAX-type coupon must be present, and it does not matter where this MAX coupon is located in the coupon list.
- If multiple coupons could be returned, and one of them is a MAX-type coupon, the system returns the MAX coupon with the highest MAX value.

Examples

1. A single MAX coupon

Cart: 2 kg apple
Period: Normal
Coupons: A5-MAX10

Price: 2 × 500 = 1000 HUF
Quantity discount: none
MAX: 10%, coupon discount: 5% → allowed
Payable: 950 HUF
Returned coupons: []

---

2. Two A5 coupons + A5-MAX10 → MAX 10% reached

Cart: 1 kg apple
Period: Normal
Coupons: A5, A5, A5-MAX10

Price: 500 HUF
First A5 → −5%
Second A5 → another −5% (total −10%, not yet validated)
A5-MAX10: MAX: 10%, another −5% → total 15%, but due to MAX 10% limit, only 10% applies
Since 10% could be reached with only one A5 + A5-MAX10, one A5 is returned
Payable: 450 HUF
Returned coupons: [A5]

---

3. A5, A5-MAX15 and A5-MAX10 → only 10% valid

Cart: 1 kg apple
Period: Normal
Coupons: A5, A5-MAX15, A5-MAX10

Price: 500 HUF
MAX values: 15% and 10% → strictest: 10%
A5 → −5%
A5-MAX15 → −5%, MAX limit 15%
A5-MAX10 → −5%, MAX limit = min(15%, 10%)
Payable: 450 HUF
Returned coupons: [A5-MAX15] (its 5% was not needed to reach 10%, and it is considered the more valuable coupon)

---

4. MAX15 and MAX10 → only 10% valid

Cart: 1 kg apple
Period: Normal
Coupons: A5-MAX15, A5-MAX10

Price: 500 HUF
MAX values: 15% and 10% → strictest: 10%
Result: 450 HUF
Returned coupons: []

---

5. MAX15 and one A5 → MAX not fully used

Cart: 1 kg apple
Period: Normal
Coupons: A5, A5-MAX15

Price: 500 HUF
MAX: 15%
A5 → −5%
A5-MAX15 → another −5%
Still 5% possible, but no more coupons
Payable: 450 HUF
Returned coupons: []

---

6. Promotion stronger than coupons

Cart: 2 kg apple
Period: Spring (automatic 15% discount)
Coupons: A5, A5, A5-MAX15

Price: 1200 HUF
Spring discount: 15%
Coupons total: 15% → not better
Payable: 1020 HUF
Returned coupons: [A5, A5, A5-MAX15]

---

7. A5-MAX15 + A5 + A5 → max 15% reached

Cart: 1 kg apple
Period: Normal
Coupons: A5-MAX15, A5, A5

Result: 425 HUF
Returned coupons: []

---

8. A5 + A5-MAX15 + A5 → max 15% reached

Cart: 1 kg apple
Period: Normal
Coupons: A5, A5-MAX15, A5

Result: 425 HUF
Returned coupons: []

---

9. A5 + A5 + A5-MAX15 → max 15% reached

Cart: 1 kg apple
Period: Normal
Coupons: A5, A5, A5-MAX15

Result: 425 HUF
Returned coupons: []

Development Requirements

The system must support the following:

- MAX coupons must be handled separately with combination logic
- The position of MAX coupons in the list does not matter
- The system must track:
    - the applied discount for the product so far
    - the given MAX limit, which cannot be exceeded
- If a coupon cannot be applied (e.g., it would exceed the MAX), it must be returned

---

Note

The introduction of MAX coupons requires complex rules and well demonstrates the increasing complexity of coupon logic.

To clarify further, the examples above are also provided in the form of unit tests:

Java Unit Tests

```java
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import java.util.List;
import org.store.*;

class StoreCR4Tests {
    static Store target;
    static Period normal;
    @BeforeAll
    public static void initStore() {
        target = new Store();
        normal = new Period("Normal");
        normal.setUnitPrice(Product.APPLE, 500.0);
        normal.setUnitPrice(Product.BANANA, 450.0);
        normal.setDiscount(Product.APPLE, 5.0, 0.1);
        normal.setDiscount(Product.APPLE, 20.0, 0.15);
        normal.setDiscount(Product.BANANA, 2.0, 0.1);
        target.addPeriod(normal);
    }
    @Test
    void test_cr4_example1_oneMAX() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 2.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5-
MAX10"));
        assertEquals(950.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example2_twoA5andMAX10() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"A5", "A5-MAX10"));
        assertEquals(450.0, price.getPrice(), 0.001);
        assertEquals(List.of("A5"), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example3_A5_MAX15_and_MAX10() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"A5-MAX15", "A5-MAX10"));
        assertEquals(450.0, price.getPrice(), 0.001);
        assertEquals(List.of("A5-MAX15"), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example4_MAX15_and_MAX10() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5-
MAX15", "A5-MAX10"));
        assertEquals(450.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example5_MAX15_and_A5() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"A5-MAX15"));
        assertEquals(450.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example6_promotionStrongerThanCoupon() {
        Period spring = new Period("Spring");
        spring.setUnitPrice(Product.APPLE, 600.0);
        spring.setDiscount(Product.APPLE, 2.0, 0.15);
        spring.setDiscount(Product.APPLE, 5.0, 0.20);
        target.addPeriod(spring);
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 2.0)));
        PriceInfo price = target.getCartPrice(cart, spring, List.of("A5",
"A5", "A5-MAX15"));
        assertEquals(1020.0, price.getPrice(), 0.001);
        assertEquals(List.of("A5", "A5", "A5-MAX15"),
price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example7_MAX15_and_twoA5() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5-
MAX15", "A5", "A5"));
        assertEquals(425.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example8_A5_MAX15_A5() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"A5-MAX15", "A5"));
        assertEquals(425.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
    @Test
    void test_cr4_example9_twoA5_and_MAX15() {
        Cart cart = new Cart(List.of(new Item(Product.APPLE, 1.0)));
        PriceInfo price = target.getCartPrice(cart, normal, List.of("A5",
"A5", "A5-MAX15"));
        assertEquals(425.0, price.getPrice(), 0.001);
        assertEquals(List.of(), price.getUnusedCoupons());
    }
}
```