Extended Euclids Algorithm

**Example 1:**

GCD(15, 69) = 3  
We can also say:  
3 = **15** \* Something which can be -9 + **69** \* Something which can be 2

(15\*60) + (69 \*(-13)) = 3  
(15\*129) + (69 \*(-28)) = 3

So there is no unique way, But the point here is that GCD can be written as a linear combination,  
  
Bezout Coefficients says  
15a + 69b   
 GCD  
  
Bezout Coefficients are the values that we multiple with m and n  
  
  
**Example 2:**

GCD(11,7) = 1  
11\* (n1) + 7\* (n2) = 1  
  
We need to first the two numbers, can be negative too  
11\* (2) + 7\* (-3) = 1  
11\* (9) + 7\* (-14) = 1  
  
  
Now we will extend the Euclid’s Algorithm which means doing the Euclid's algorithm, we will be able to calculate the Bezout coefficients as I calculate the GCD, that's called the extended Euclid's algorithm.

First the normal Euclid’s Algo:  
A person writing on a glass board

Description automatically generated

The same operations that we are doing a loop of Euclid’s Algo but at the same time in a new column we operate the Bezout coefficients

1. A person writing on a blackboard

   Description automatically generated

A person looking at a math equation

Description automatically generated  
In the second picture we see below, a screenshot is also taken below:  
A hand writing on a blackboard

Description automatically generated  
Here the idea is not only we calculate the Euclid’s Algo but at the same time in a new column we operate the Bezout coefficients, using different notations. With normal alphabets its for the Euclid’ Algo, whereas for the Alphabets that have hat on top of them, they represent the Bezout Coefficients

**Example 3 – Summarizing All of the Concept:**GCD (54, 12)  
A person writing on a blackboard

Description automatically generated

**Extended Euclidean Algorithm: GCD and Bézout Coefficients**

# **Example: GCD Calculation and Bézout Coefficients**

Problem Statement:  
Calculate the GCD of 54 and 12 using the extended Euclidean algorithm and find the corresponding Bézout coefficients.

## Steps:

1. Initialize Variables:  
 - Input values: m₀ = 54, n₀ = 12  
 - Bézout coefficients: s = 1, t = 0  
 - Secondary coefficients: ŝ = 0, t̂ = 1

2. First Step (Division: 54 ÷ 12):  
 - Quotient (q) = 4  
 - Remainder (r) = 6  
 - Update values: m = 12, n = 6  
 - New Bézout coefficients: s = 0, t = 1  
 - New secondary coefficients: ŝ = 1, t̂ = -4  
 - Expression: 6 = 54 - 4 × 12

3. Second Step (Division: 12 ÷ 6):  
 - Quotient (q) = 2  
 - Remainder (r) = 0  
 - Update values: m = 6, n = 0  
 - New Bézout coefficients: s = 1, t = -4  
 - New secondary coefficients: ŝ = -2, t̂ = 9  
 - Expression: 6 = 1 × 54 - 4 × 12

4. Result:  
 - GCD = 6  
 - Bézout Coefficients: s = 1, t = -4  
 - Final Expression: 6 = 1 × 54 - 4 × 12

Summary:  
Using the extended Euclidean algorithm, the GCD of 54 and 12 is 6, with Bézout coefficients s = 1 and t = -4.

# **Generalized Process for the Extended Euclidean Algorithm**

## Steps:

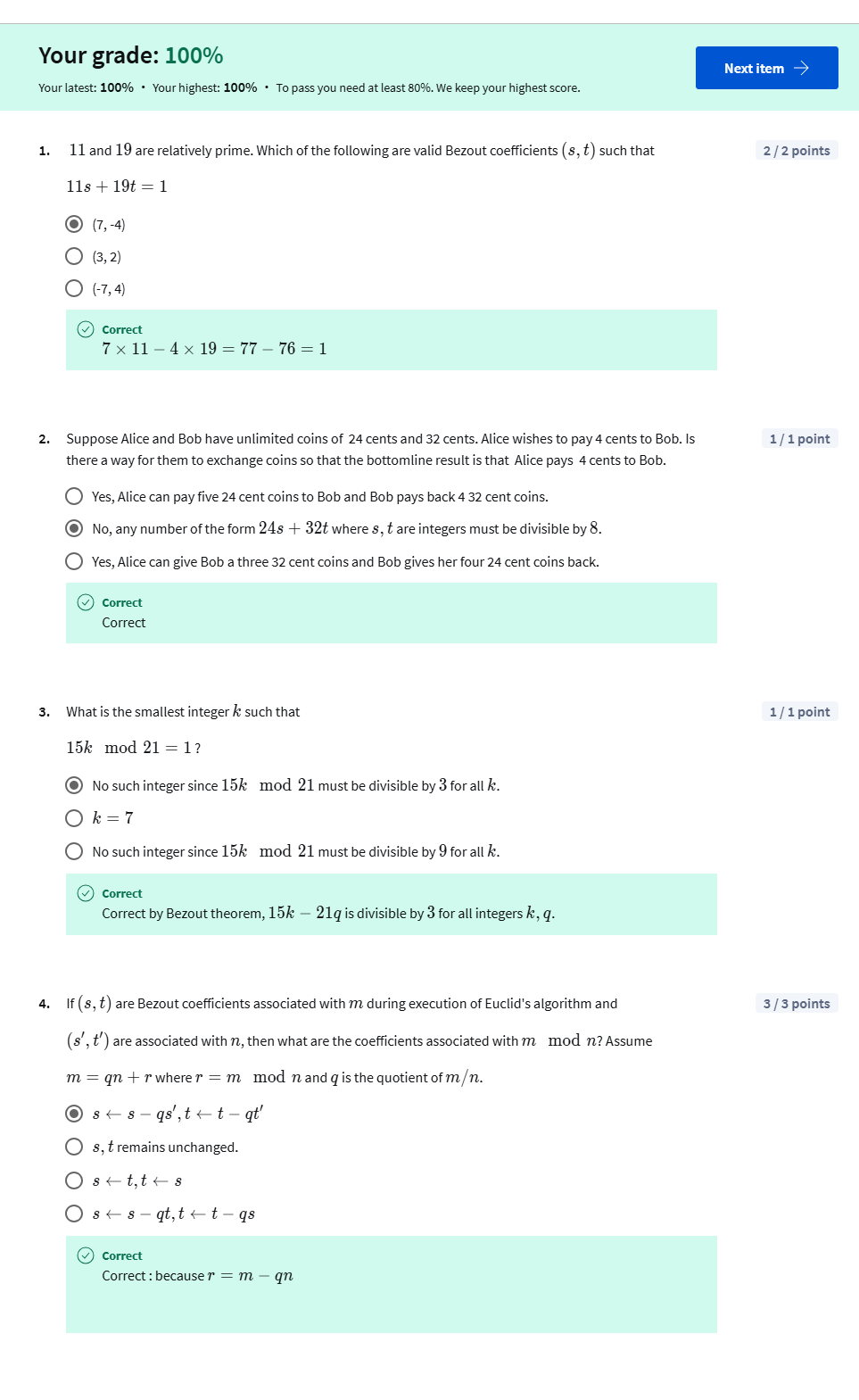
1. Initialize Variables:  
 - Set the two numbers: a (larger number), b (smaller number)  
 - Initialize Bézout coefficients: s₀ = 1, t₀ = 0, s₁ = 0, t₁ = 1

2. Repeat the Division Process:  
 - Loop until remainder is 0:  
 a. Calculate Quotient and Remainder:  
 - q = a ÷ b  
 - r = a mod b  
 - Update: a = b, b = r  
 b. Update Bézout Coefficients:  
 - s\_new = s₀ - q × s₁  
 - t\_new = t₀ - q × t₁  
 - Update s₀, s₁, t₀, t₁ for the next iteration

3. End Condition:  
 - When r = 0, the current a is the GCD.  
 - The values of s₀ and t₀ are the Bézout coefficients.

4. Result:  
 - GCD = a (the last non-zero remainder)  
 - Bézout coefficients satisfy the relationship: GCD(a, b) = s₀ × a₀ + t₀ × b₀

**Summary of Algorithm:**  
1. Initialize numbers a and b and Bézout coefficients.  
2. Perform division and update coefficients iteratively.  
3. Terminate when the remainder is 0.  
4. Return the GCD and Bézout coefficients.

**Assignment:  
**