4)
```
public int hashCode() {
    int hash = 0;
    for (int i = 0; i < length(); i++) {
        hash = (hash * 31) + charAt(i);
    }
    return hash;
}
```

a) $\square = 68$ $\quad i=0 : (0 * 31) + 68 = 68$

$d = 100$ $\quad i=1 : (68 * 31) + 100 = \underline{2208}$

$E = 69$ $\quad i=0 : (0 * 31) + 69 = 69$

$i=1 : (69 * 31) + 69 = \underline{2208}$

For any $n \geq 1$; if the sum of the characters in String A = the sum of the characters in String B and the length of A = length of B, then using the hashCode() algorithm will result in the same string of length $2n$, where n is the length of A and B. There are 2^n possible result strings of input length of n that satisfy this property.