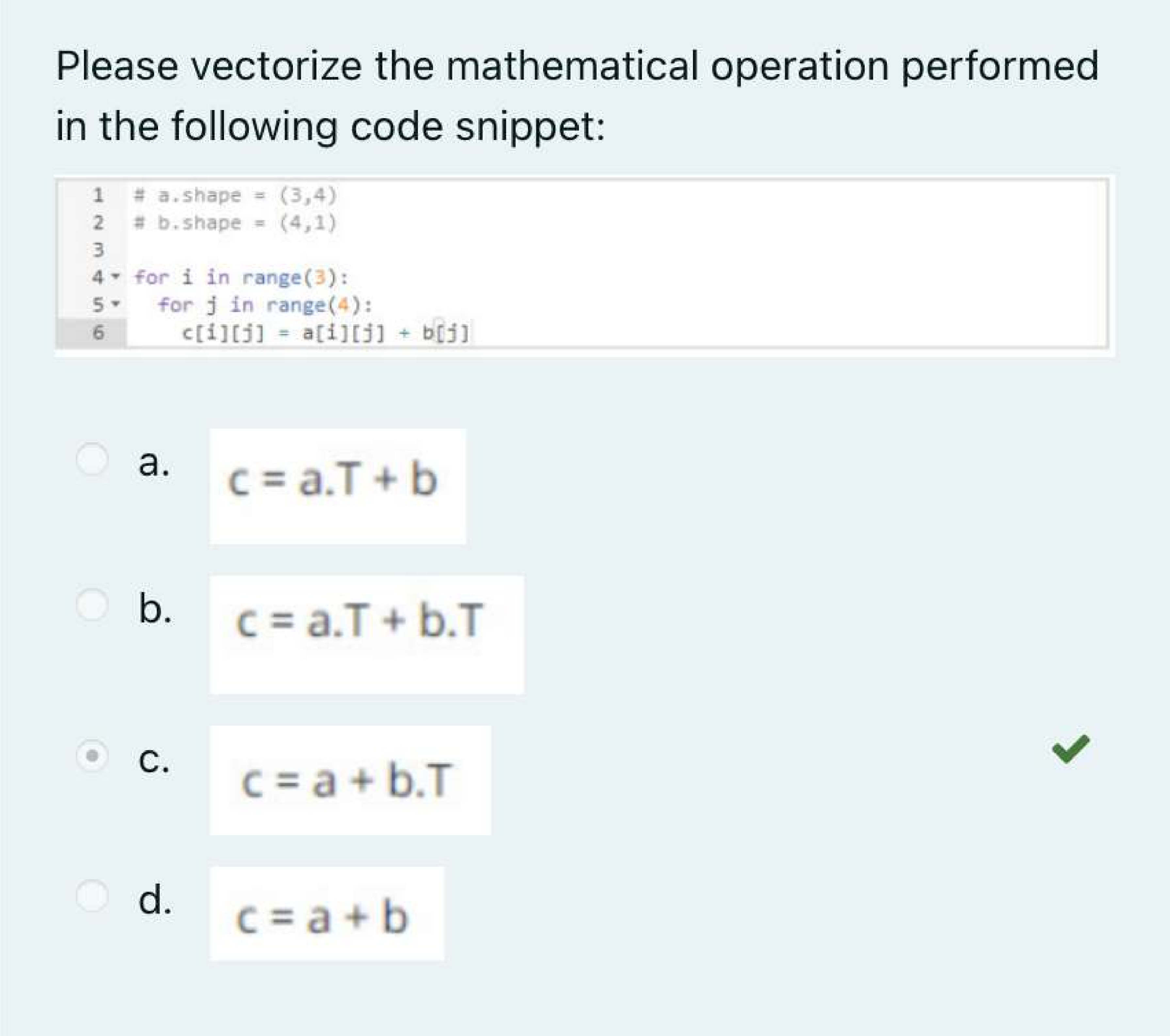


Please vectorize the mathematical operation performed in the following code snippet:

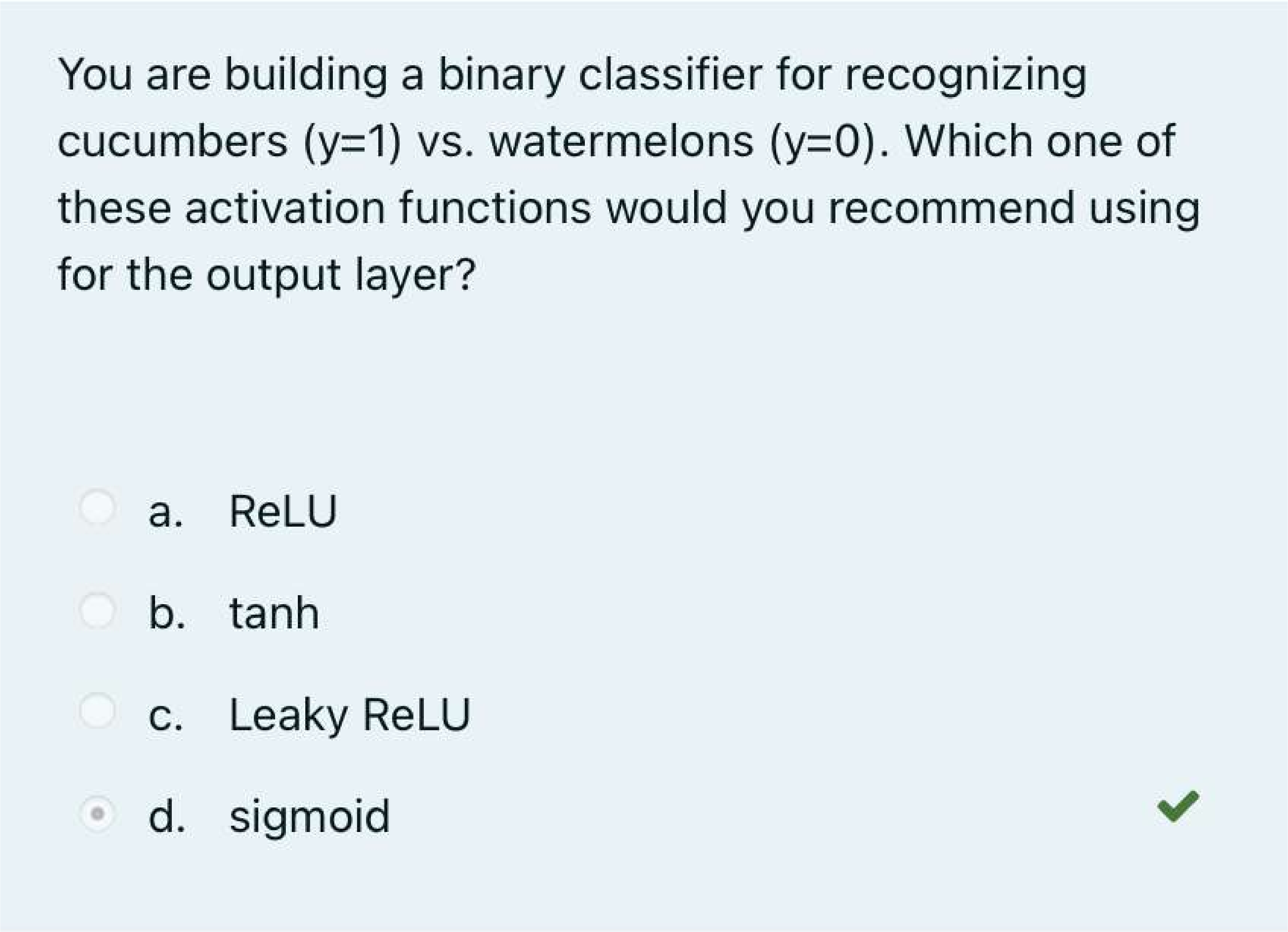
```
# a.shape = (3,4)
# b.shape = (4,1)

for i in range(3):
  for j in range(4):
    c[i][j] = a[i][j] + b[j]
```

○ a. c = a.T + b

○ b. c = a.T + b.T

● c. c = a + b.T ✓

○ d. c = a + b



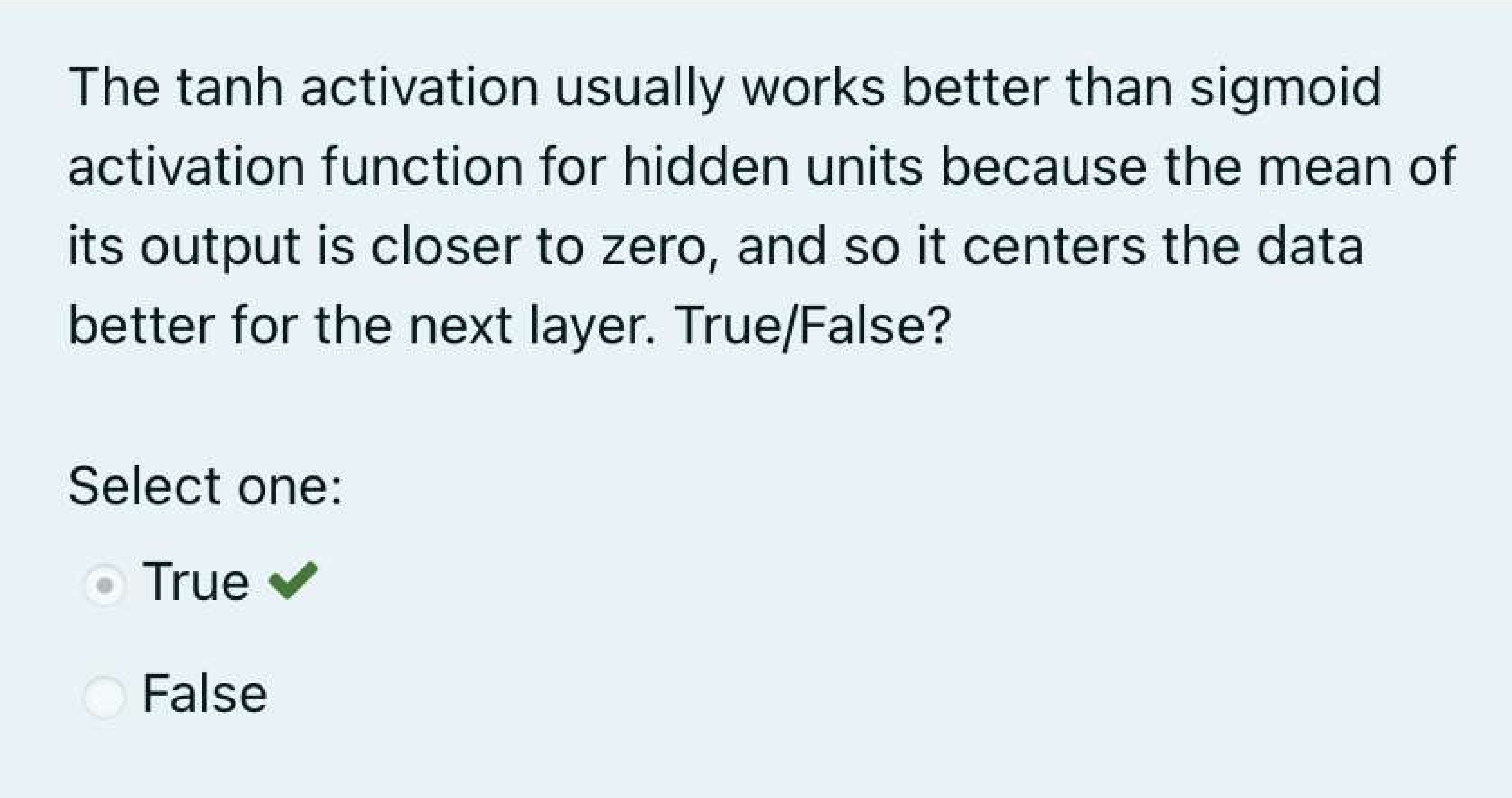
You are building a binary classifier for recognizing cucumbers (y=1) vs. watermelons (y=0). Which one of these activation functions would you recommend using for the output layer?

○ a. ReLU

○ b. tanh

○ c. Leaky ReLU

● d. sigmoid ✓

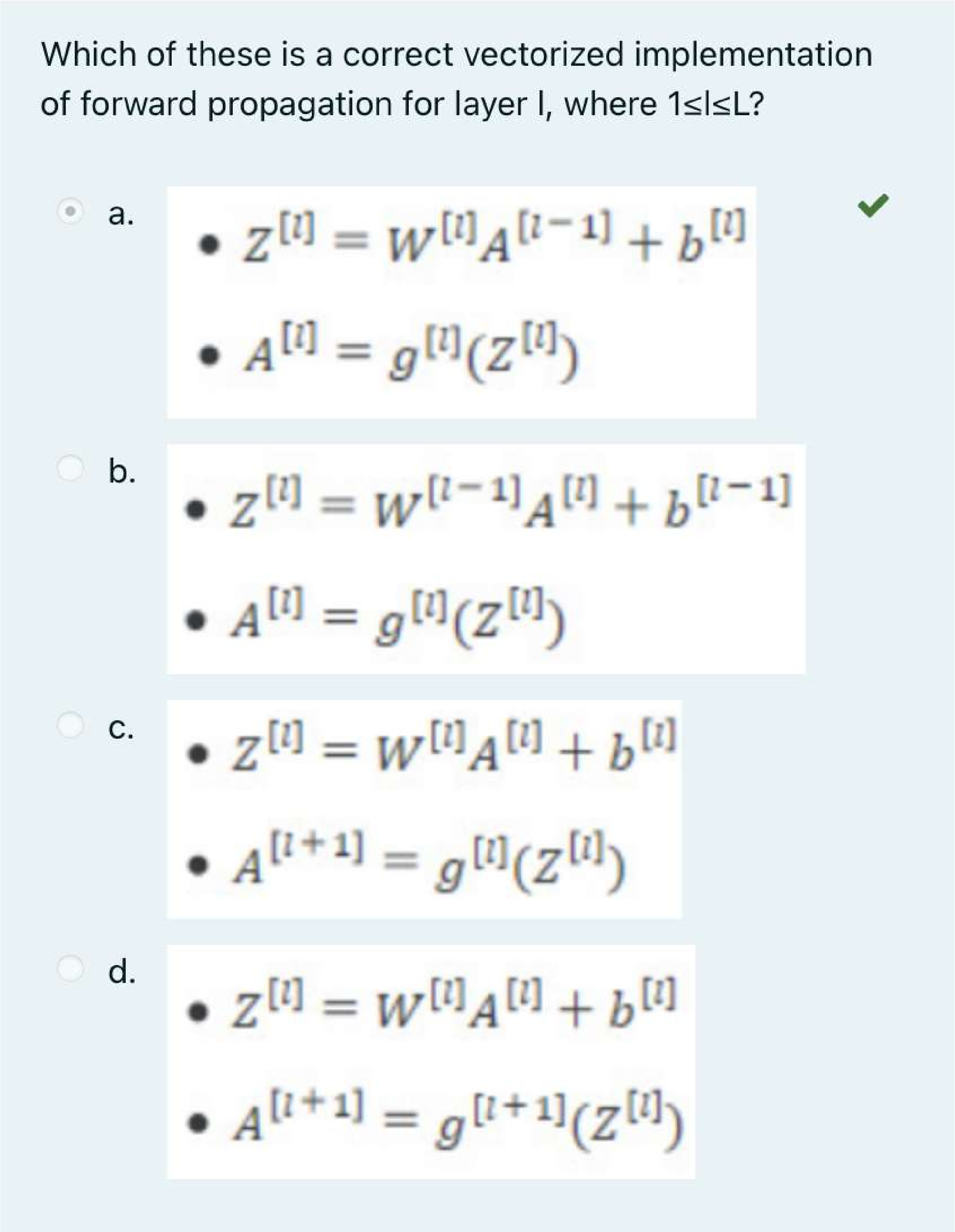


The tanh activation usually works better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data better for the next layer. True/False?

Select one:

● True ✓

○ False



Which of these is a correct vectorized implementation of forward propagation for layer l, where 1≤l≤L?

● a.

- $Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$
- $A^{[l]} = g^{[l]}(Z^{[l]})$

✓

○ b.

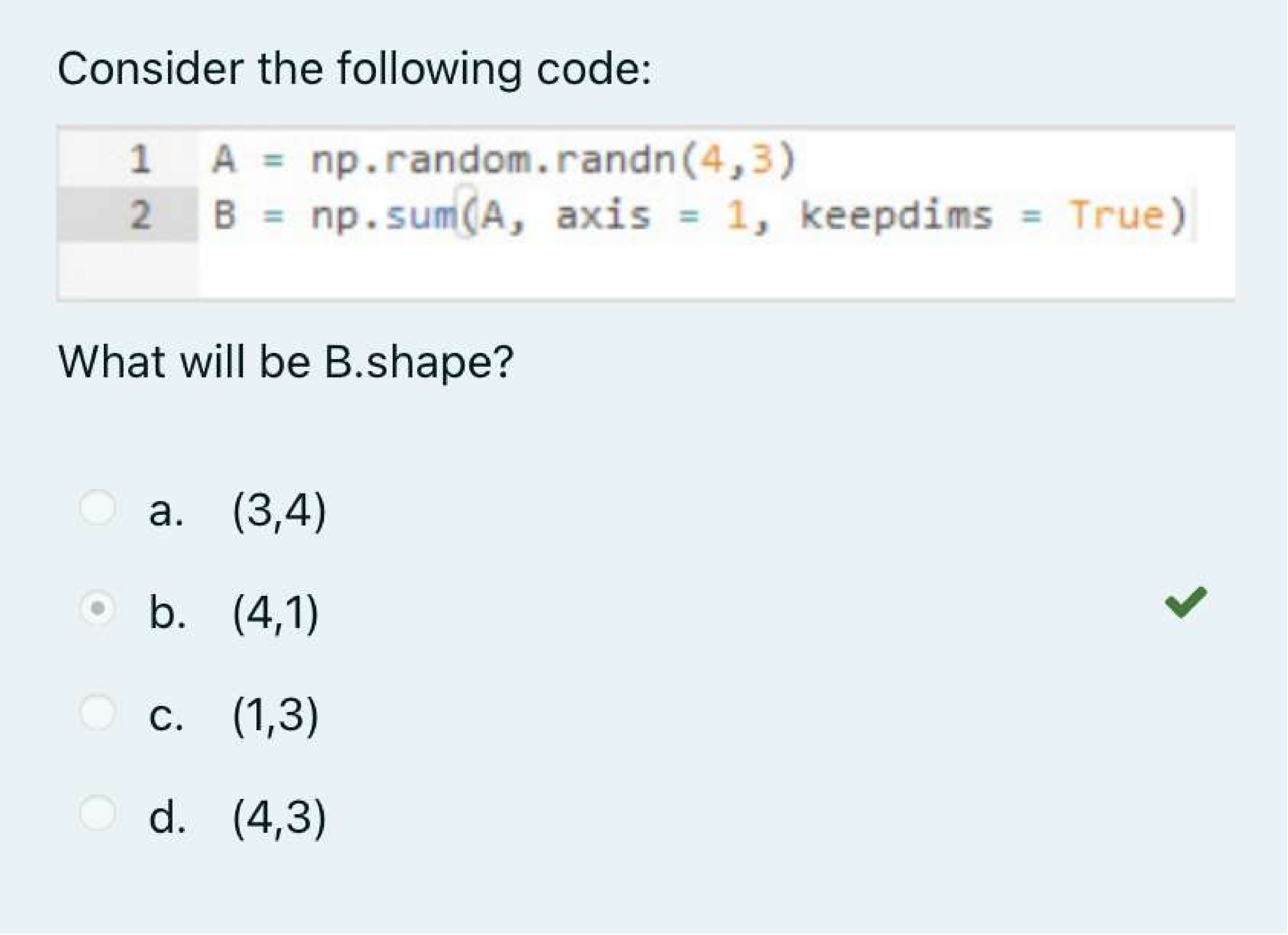
- $Z^{[l]} = W^{[l-1]}A^{[l]} + b^{[l-1]}$
- $A^{[l]} = g^{[l]}(Z^{[l]})$

○ c.

- $Z^{[l]} = W^{[l]}A^{[l]} + b^{[l]}$
- $A^{[l+1]} = g^{[l]}(Z^{[l]})$

○ d.

- $Z^{[l]} = W^{[l]}A^{[l]} + b^{[l]}$
- $A^{[l+1]} = g^{[l+1]}(Z^{[l]})$



Consider the following code:

```
A = np.random.randn(4,3)
B = np.sum(A, axis = 1, keepdims = True)
```

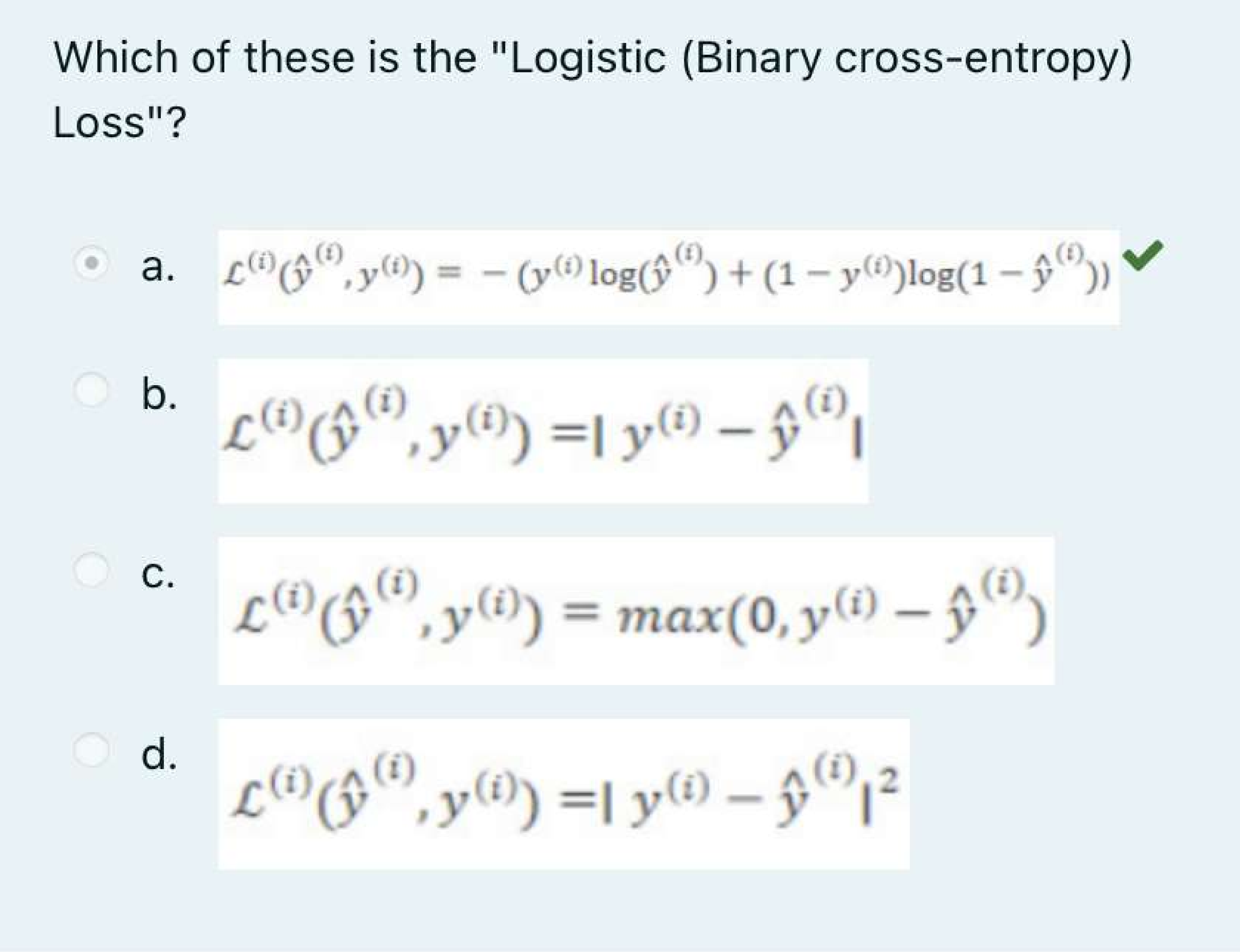
What will be B.shape?

○ a. (3,4)

● b. (4,1) ✓

○ c. (1,3)

○ d. (4,3)



Which of these is the "Logistic (Binary cross-entropy) Loss"?

● a. $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)}\log(\hat{y}^{(i)}) + (1 - y^{(i)})\log(1 - \hat{y}^{(i)}))$ ✓

○ b. $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = | y^{(i)} - \hat{y}^{(i)} |$

○ c. $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = max(0, y^{(i)} - \hat{y}^{(i)})$

○ d. $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = | y^{(i)} - \hat{y}^{(i)} |^2$