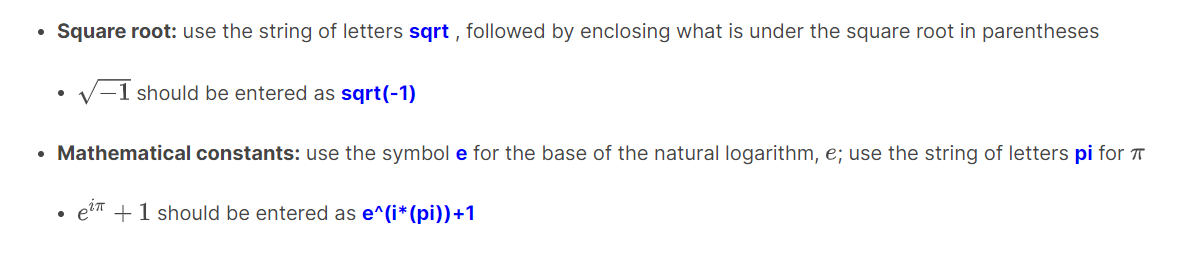
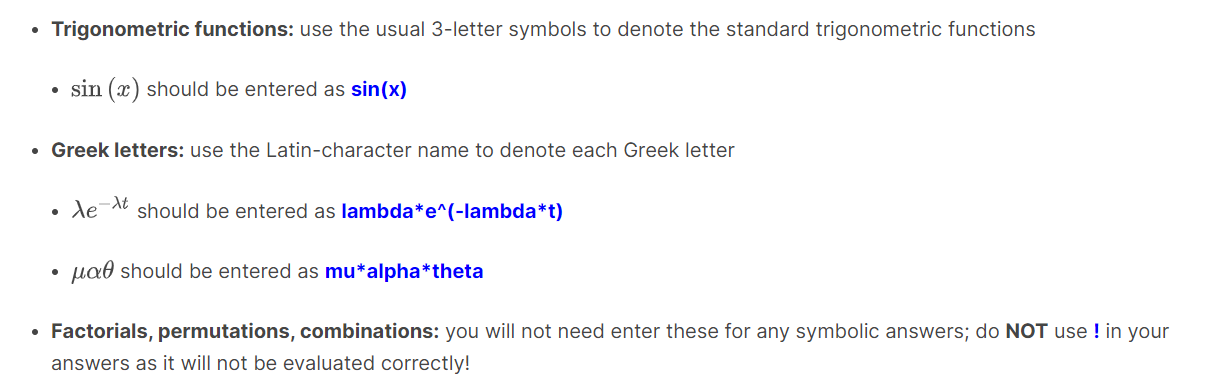
Homework mechanics and standard notation

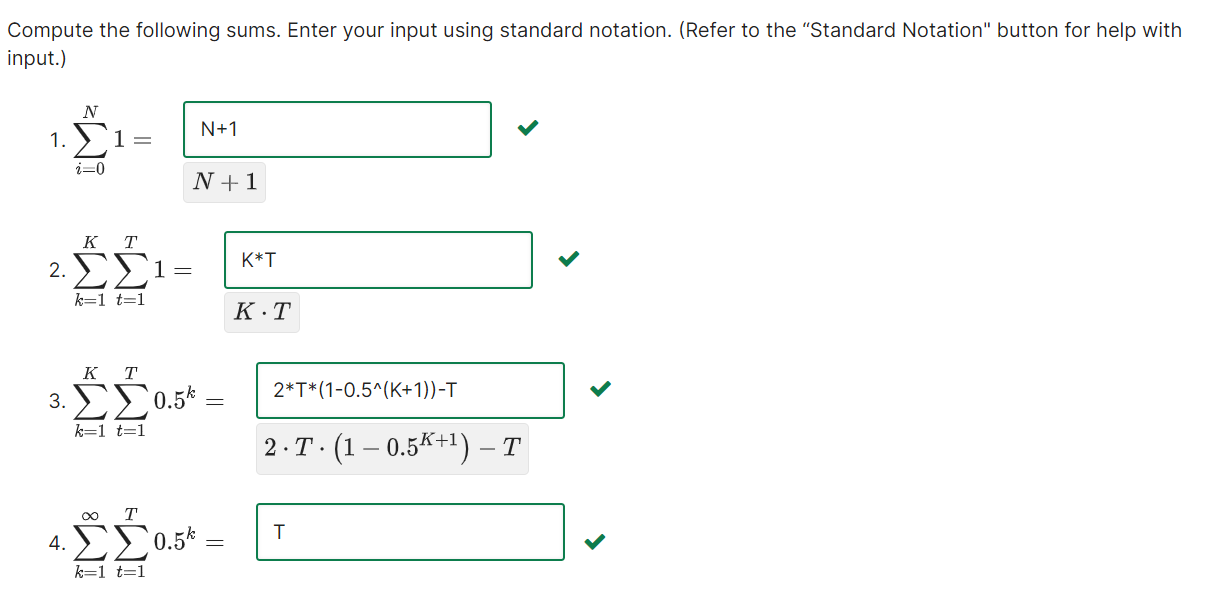




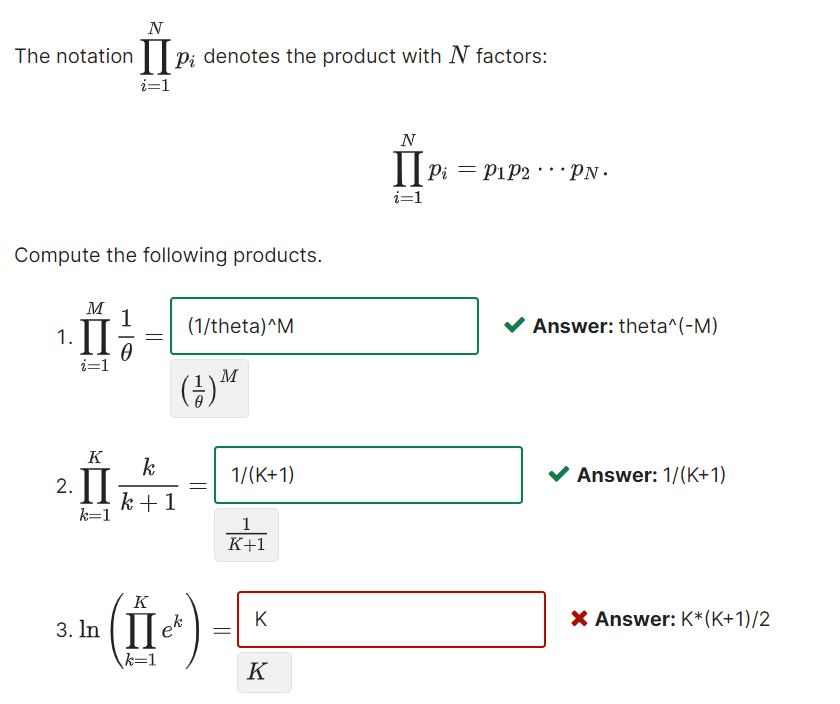
# Unit 0 Vectors, Planes, and Optimization

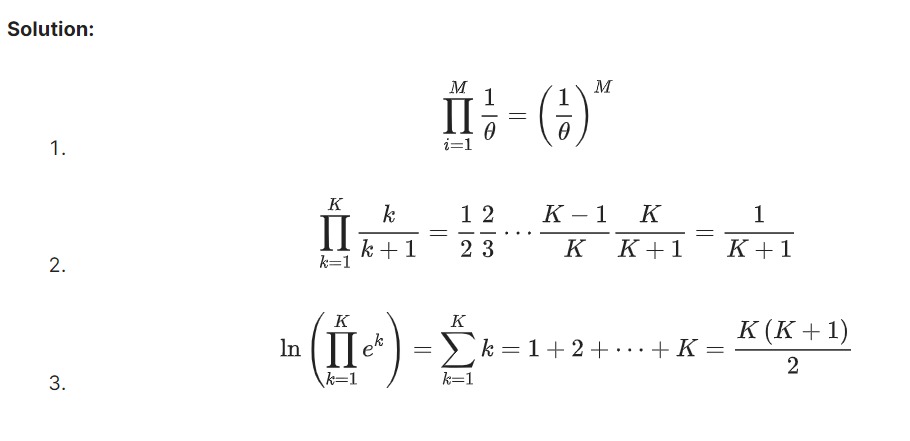
## Homework 0

### Summation Notation



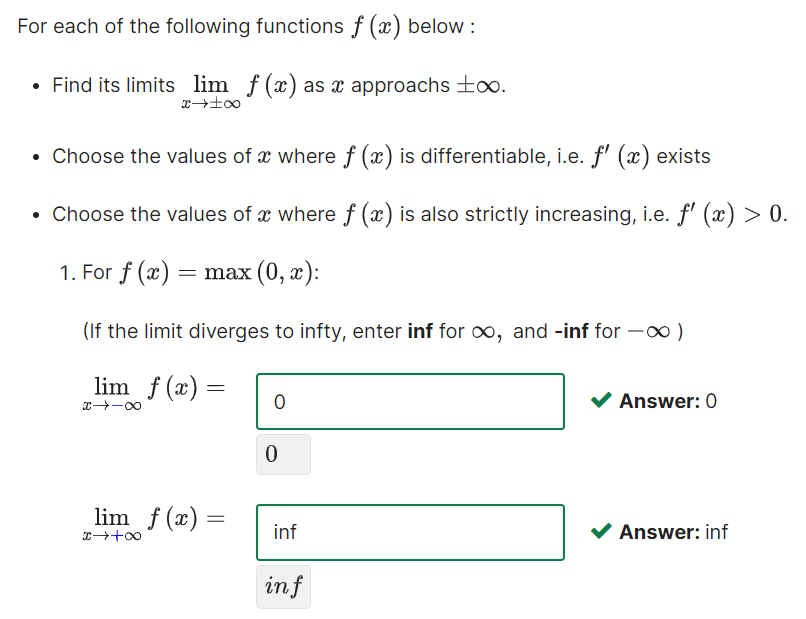
### Product Notation

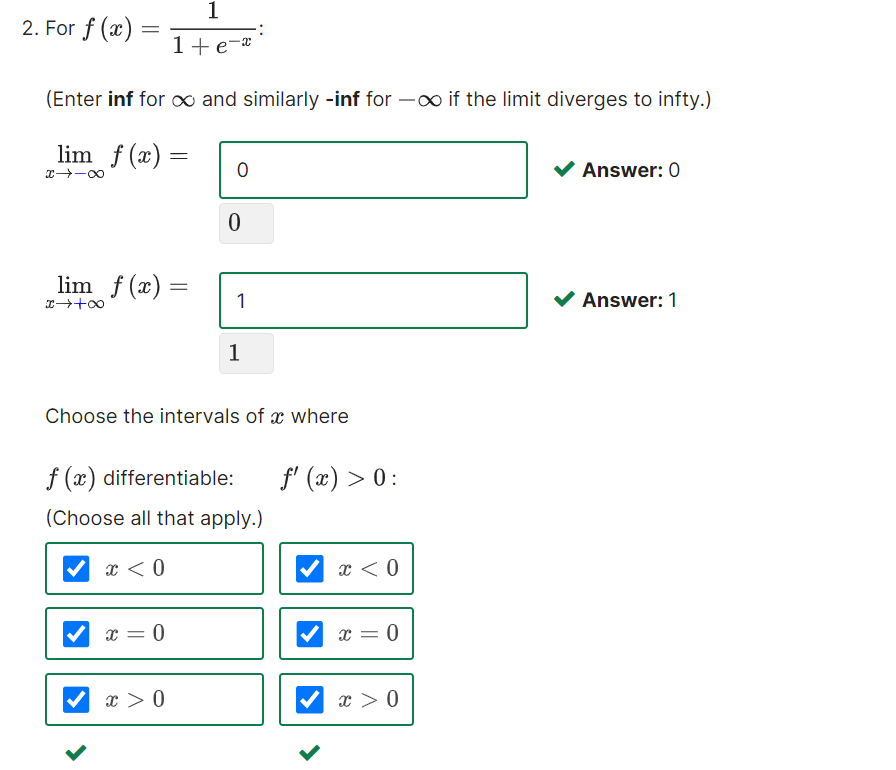
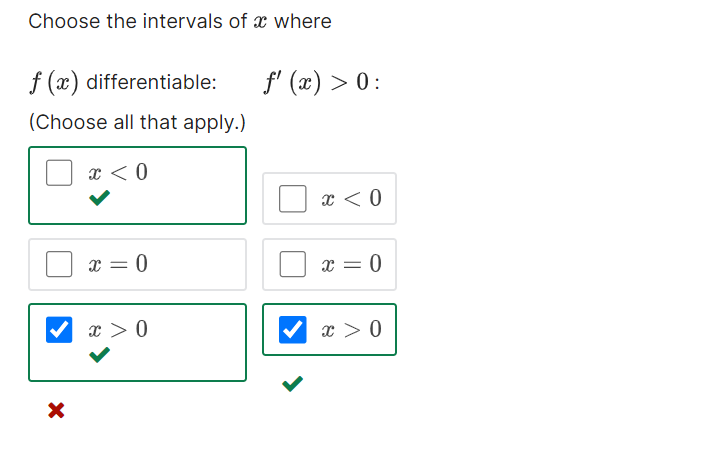


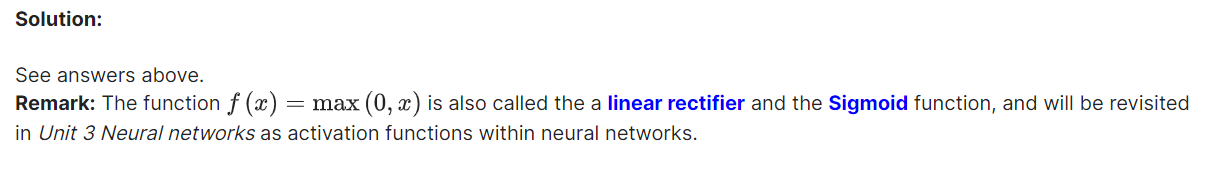


### Function Properties

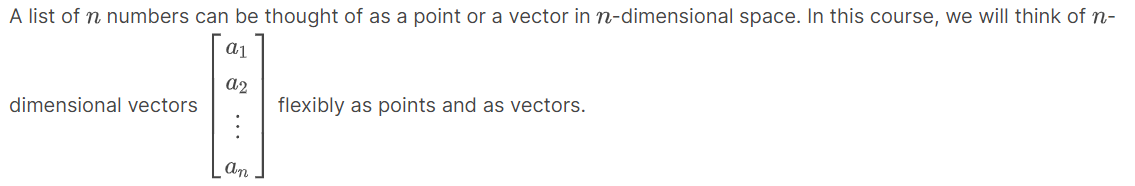
Asymptotics and Trends





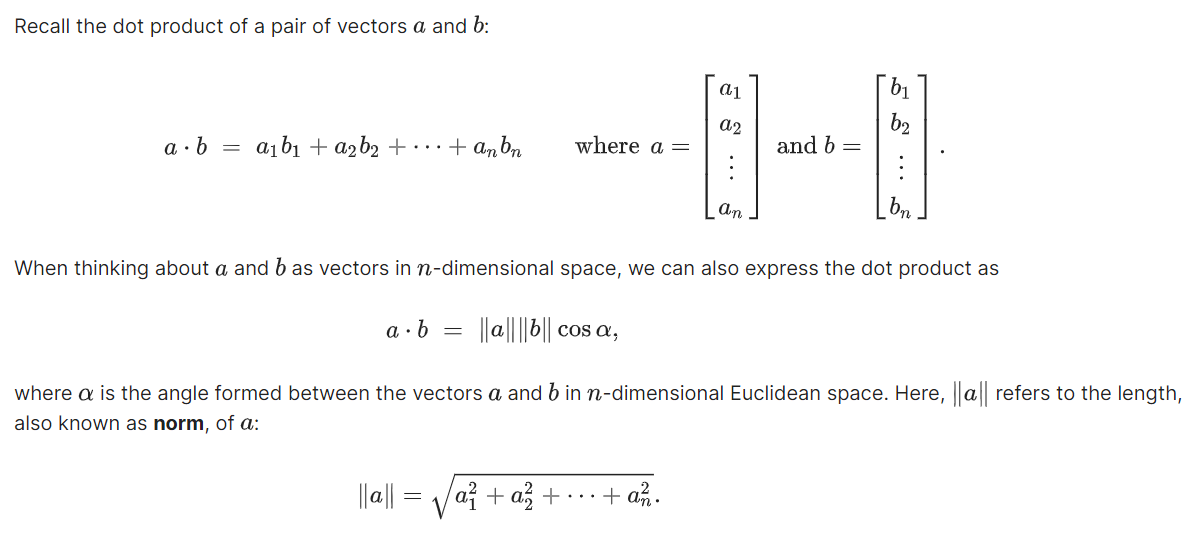


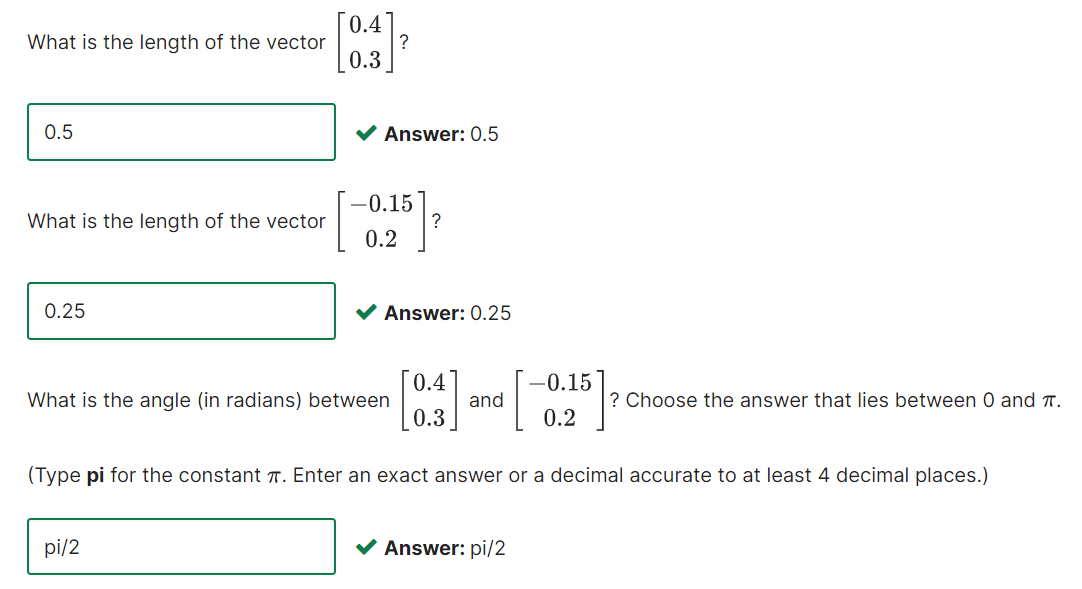
### Points and Vectors

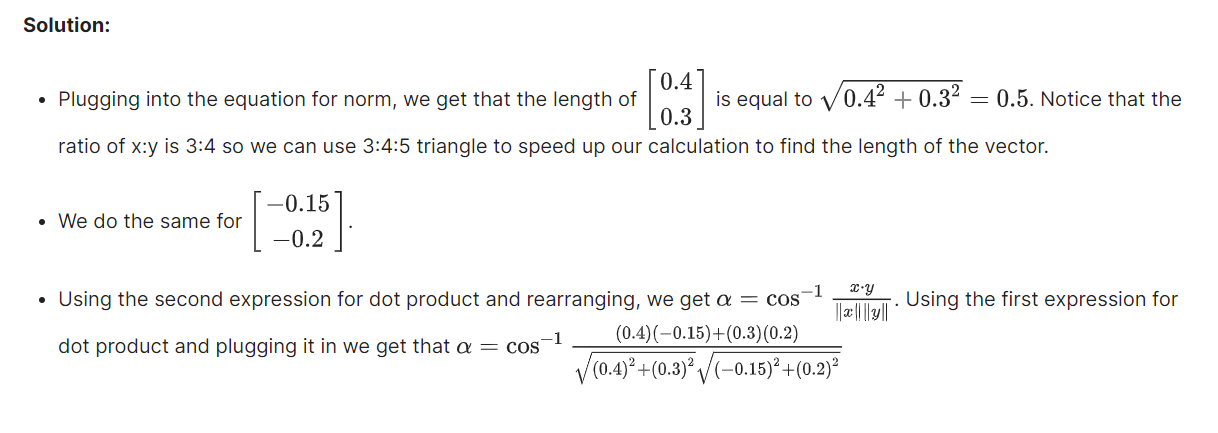


#### Dot Products and Norm – Tích vô hướng và chuẩn tắc

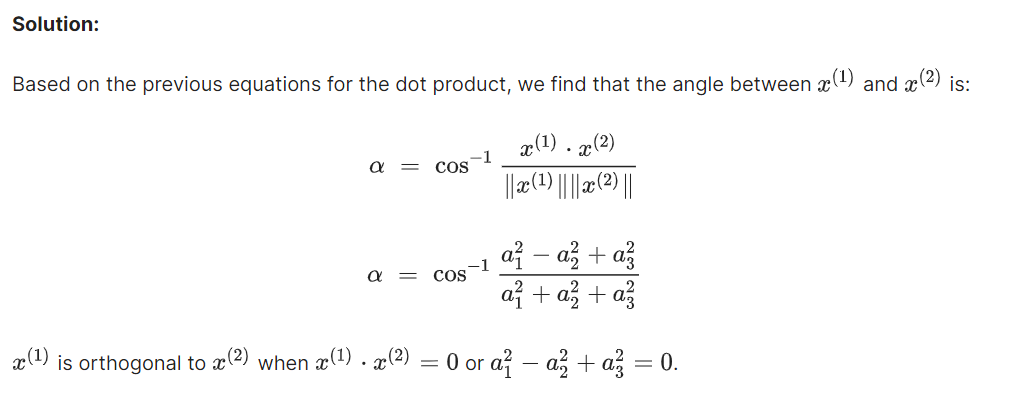
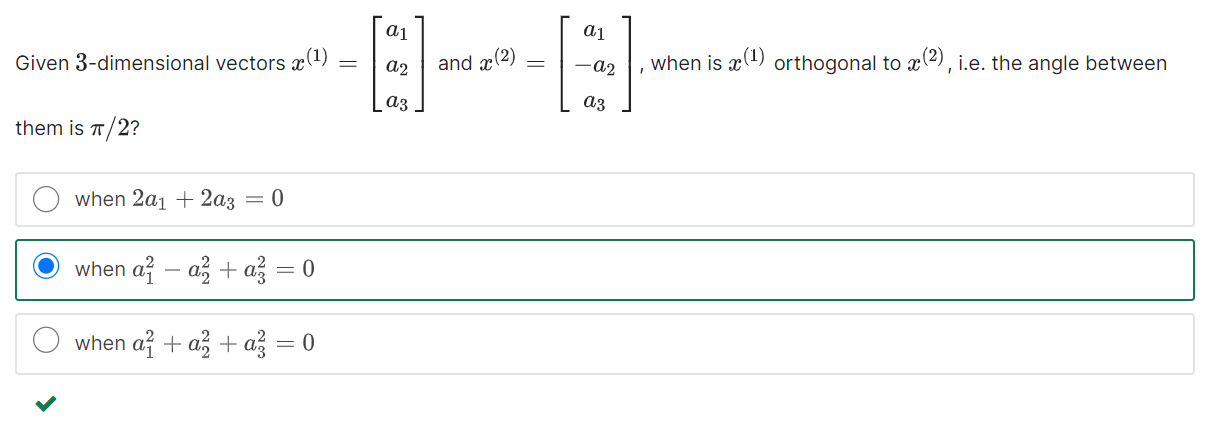
numbers, vectors, matrices, planes, hyperplanes



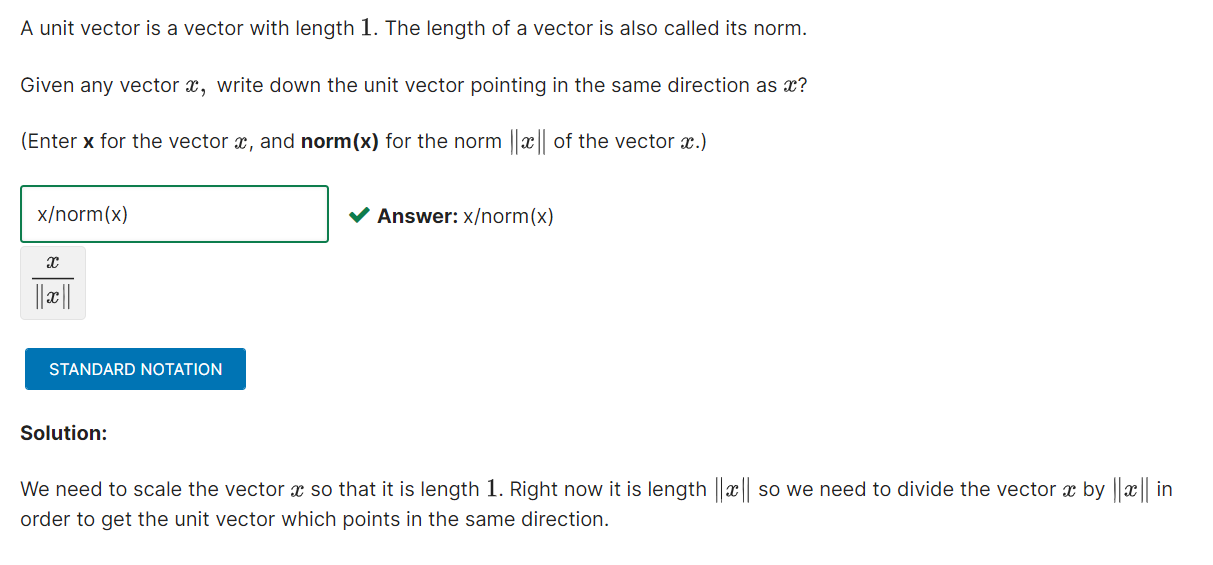




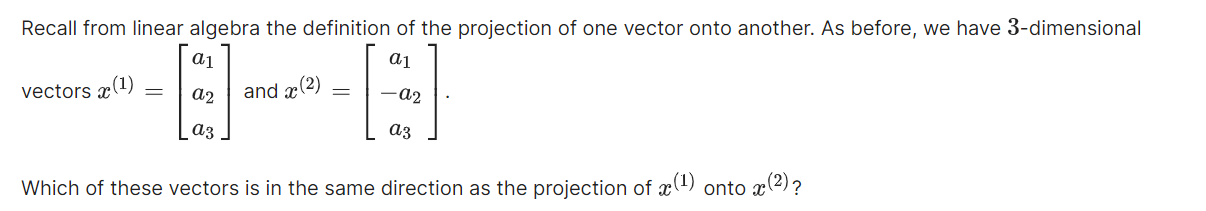
#### Dot Products and Orthogonality - Tích vô hướng và trực giao

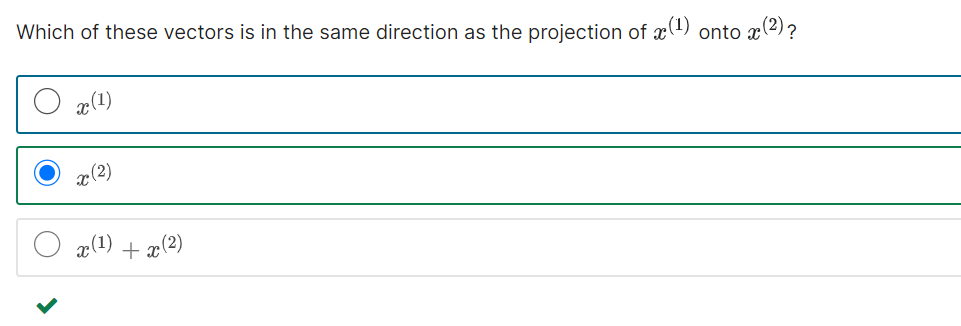


#### Unit Vectors – Vector đơn vị

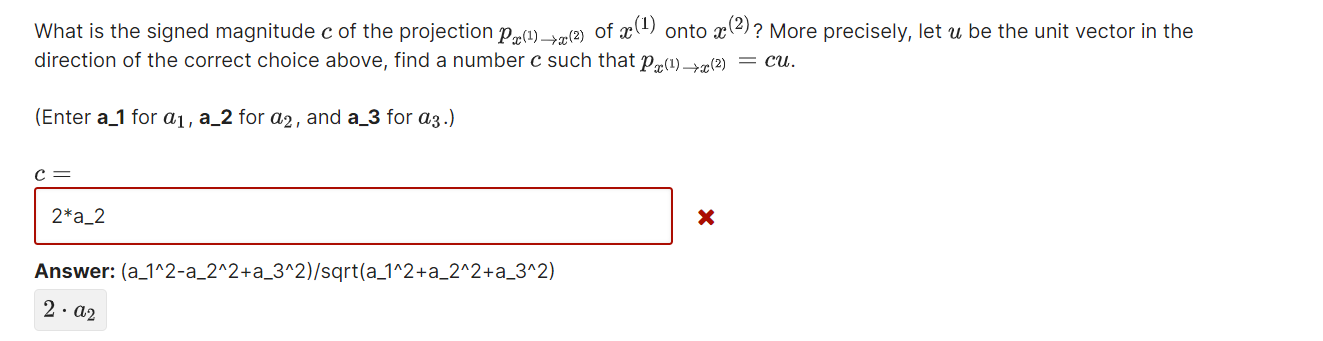


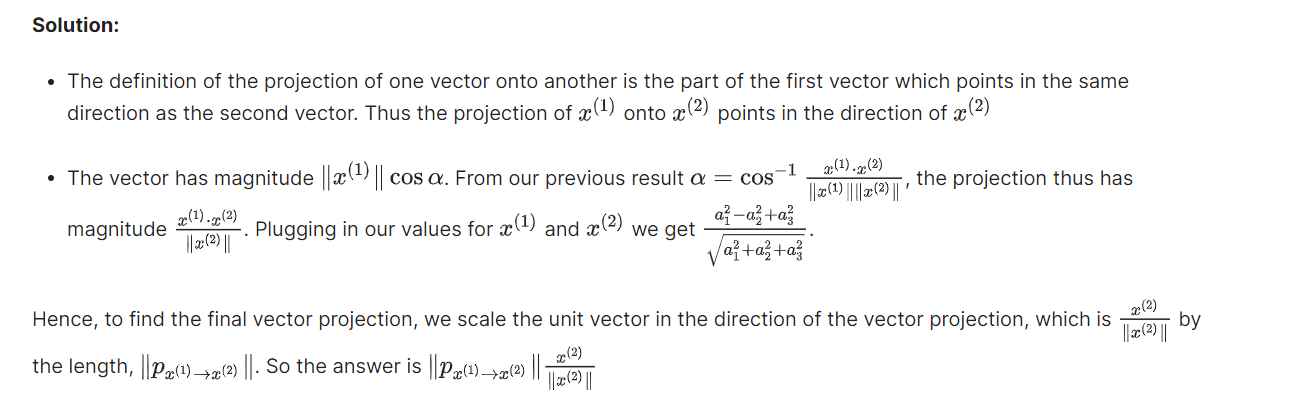
#### Projections – Phép chiếu



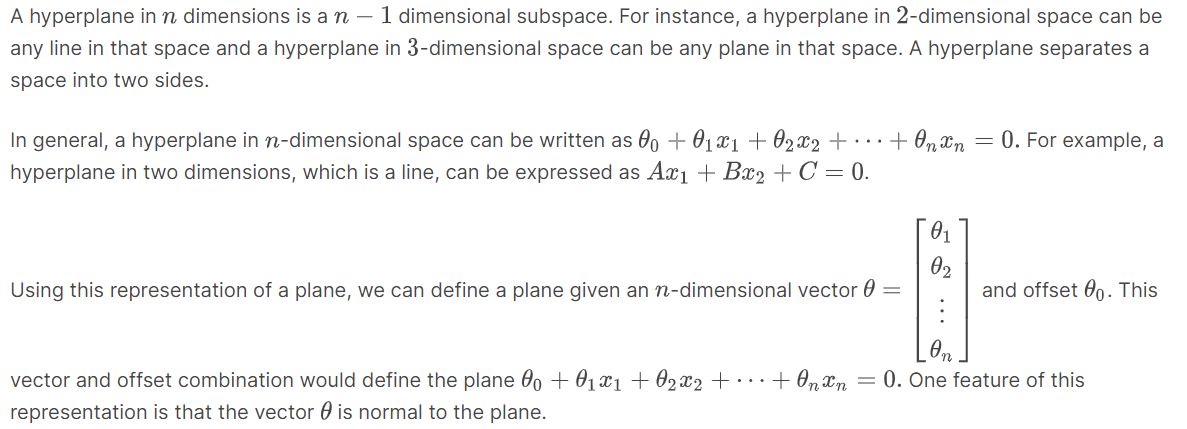


Tìm Signed magnitude c của phép chiếu vector x1 lên x2.

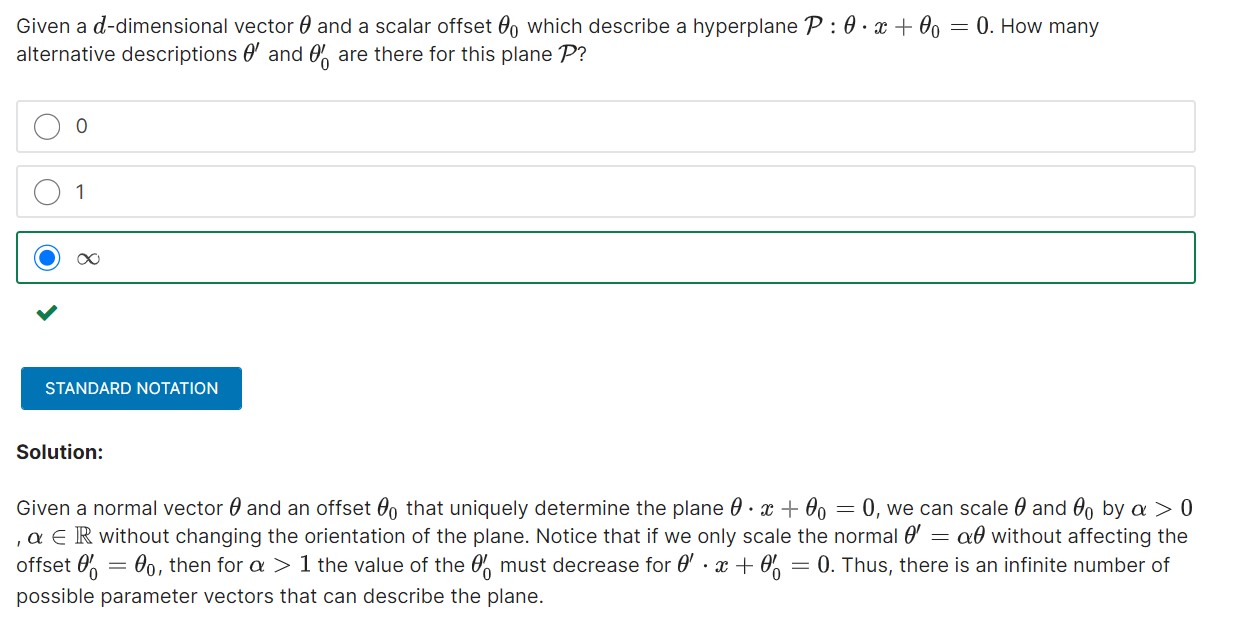




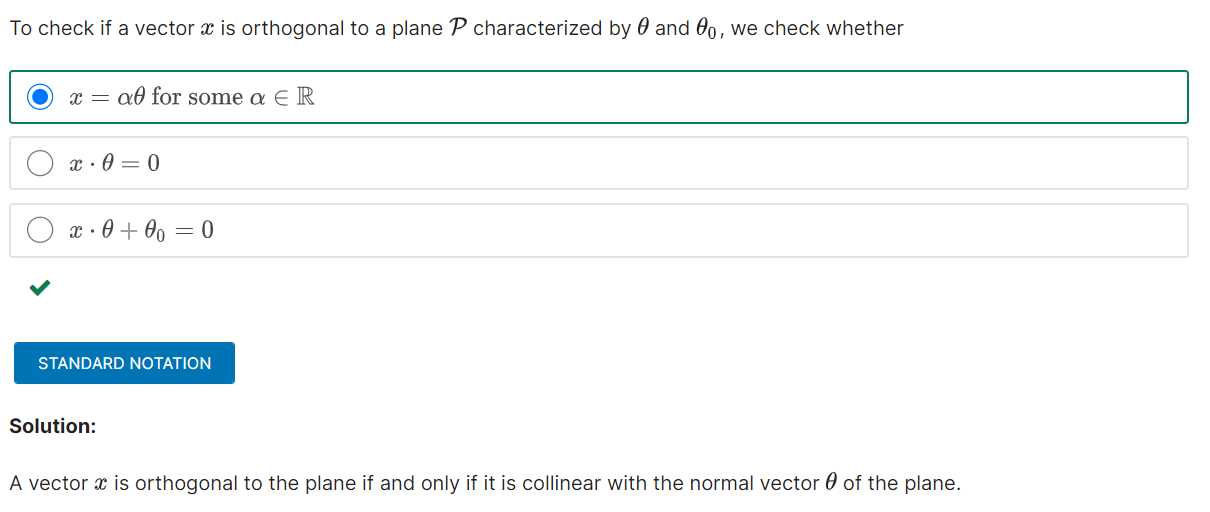
### Planes – Mặt phẳng (MP)



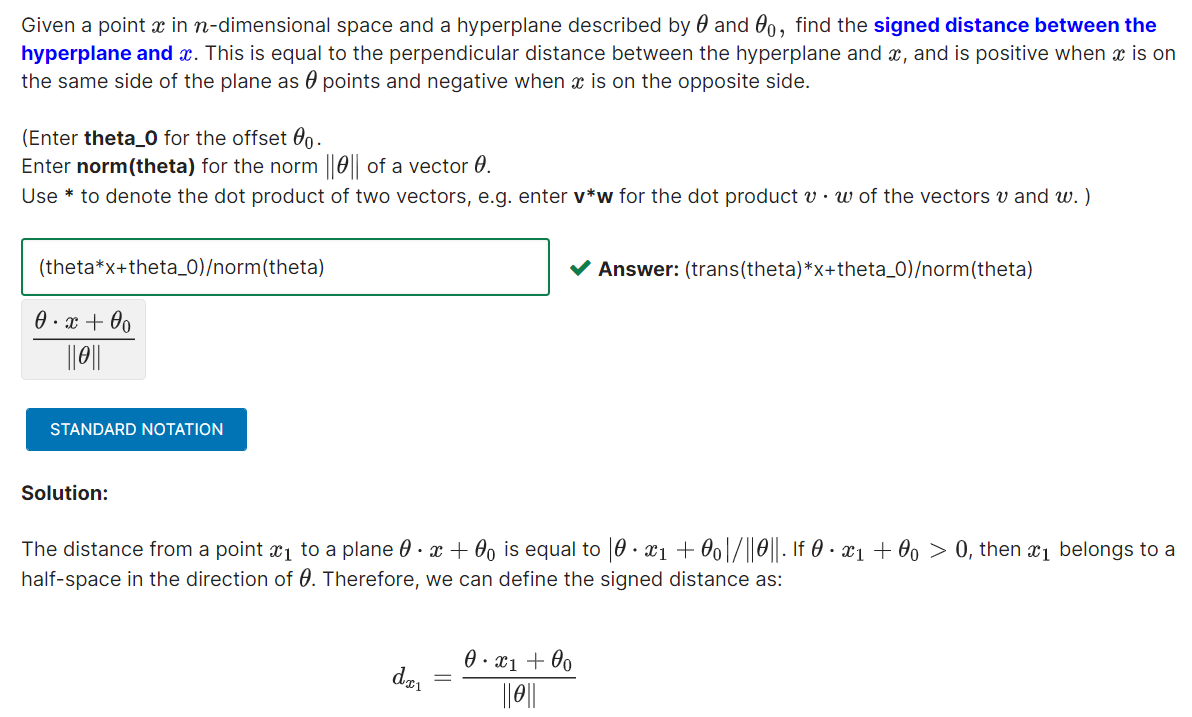
#### Number of Representations



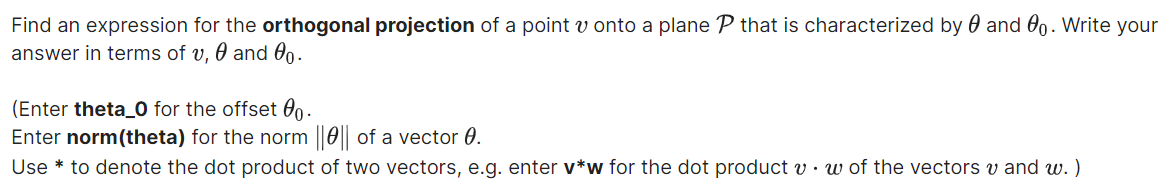
#### Orthogonality Check – Vector vuông góc mp

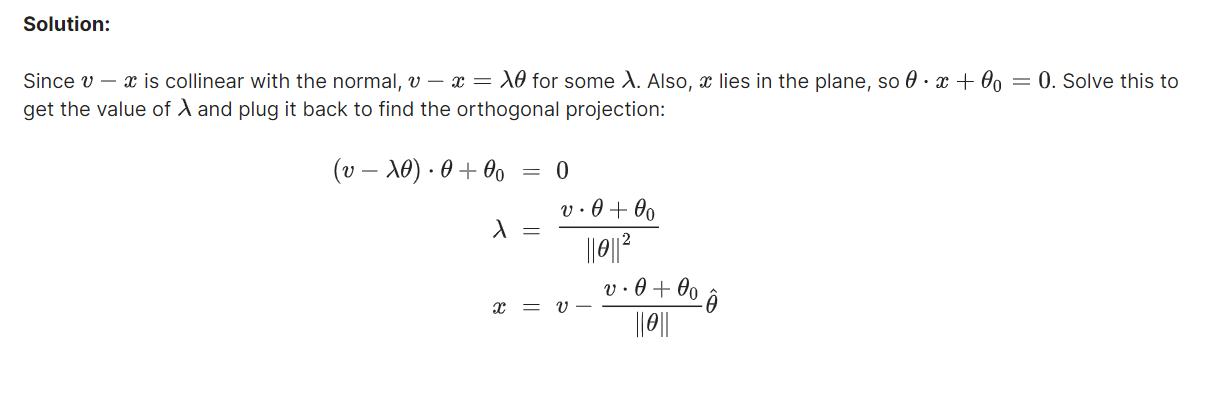


#### Perpendicular Distance to Plane – Khoản cách tới mp

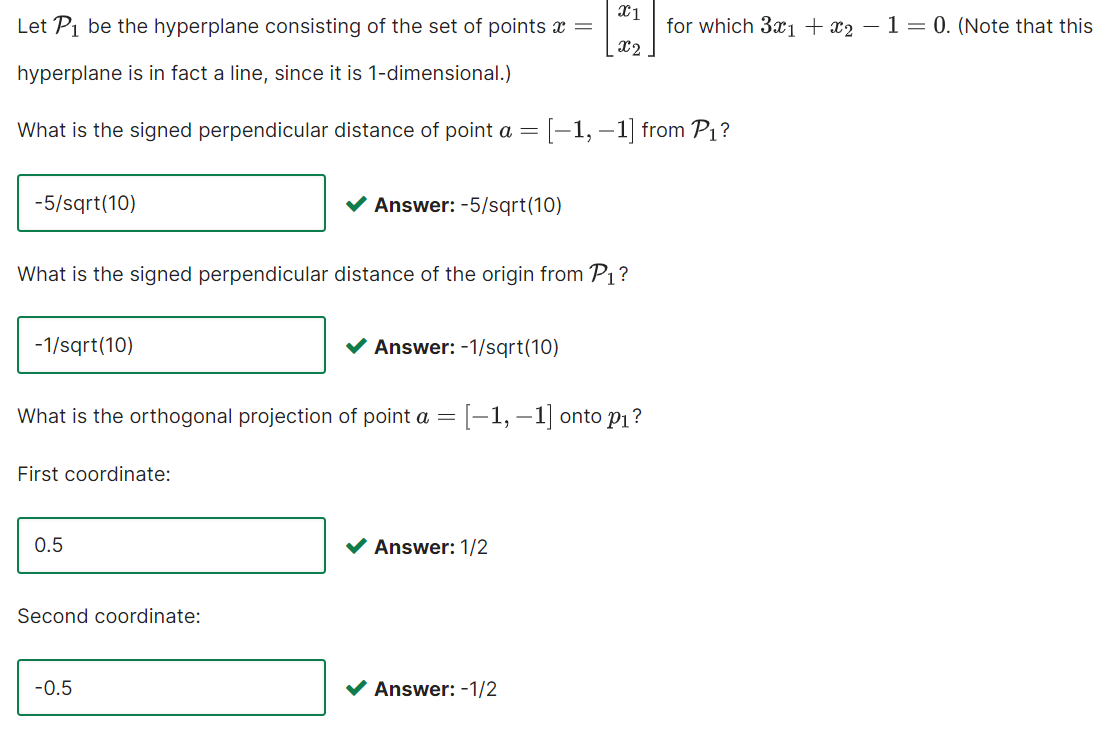


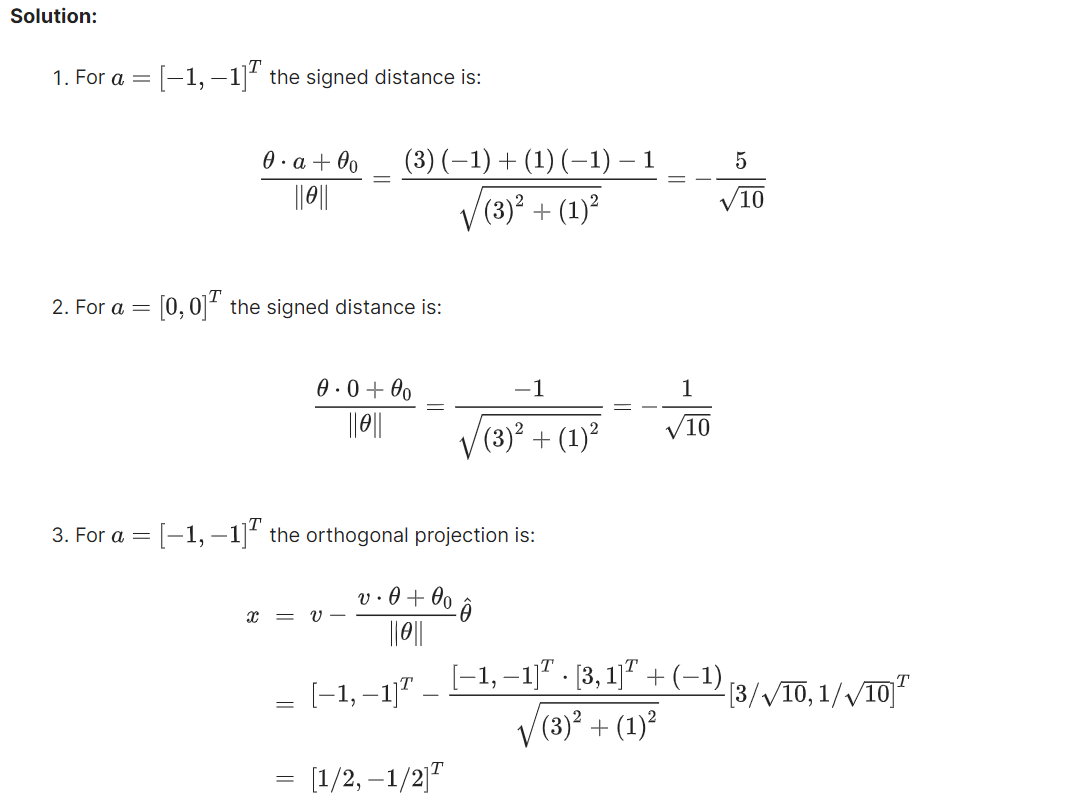
#### Orthogonal Projection onto Plane – Phép chiếu vevtor lên mp

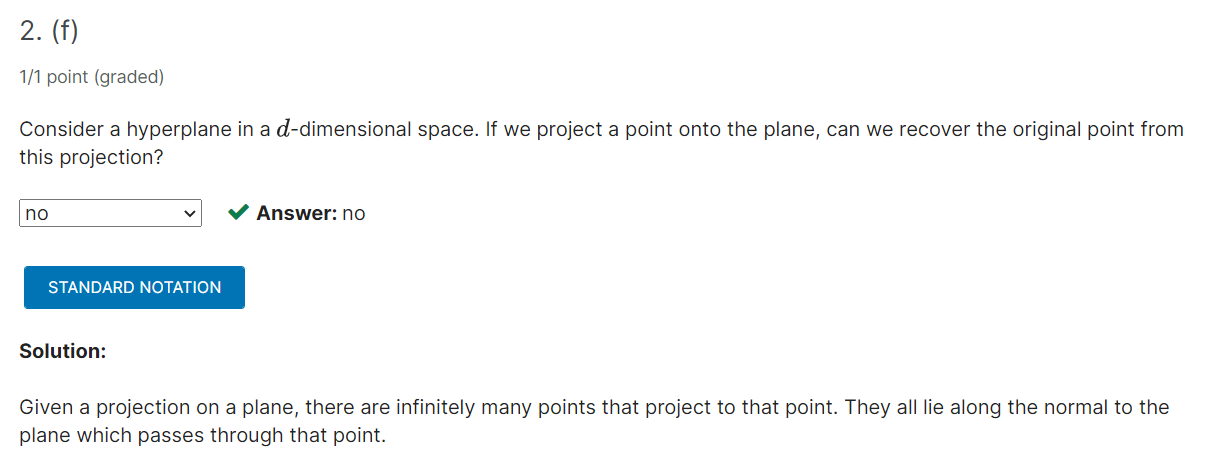
 



#### Perpendicular Distance to Plane



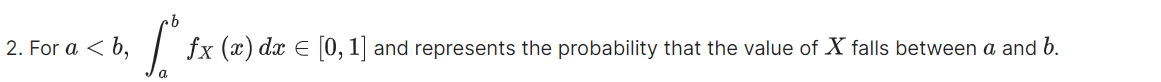


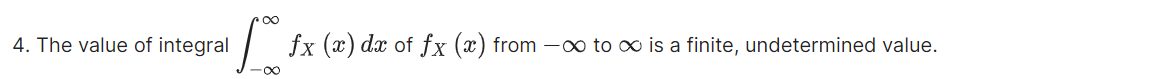


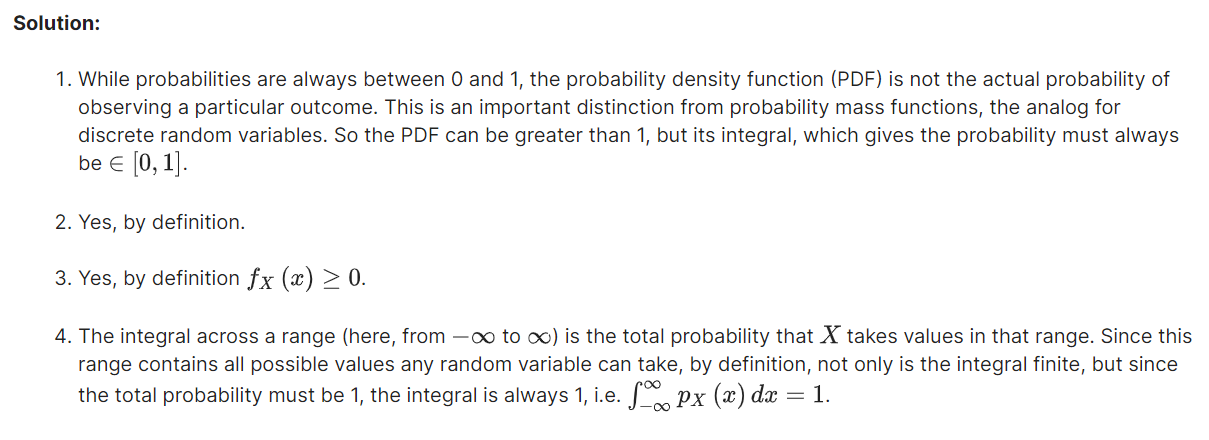
### Probability: Pro Density Functions – Hàm mật độ xác suất



*Cho X là biến thay đổi ngẫu nhiên liên tục với hàm mật độ xác suất là fx(x)*

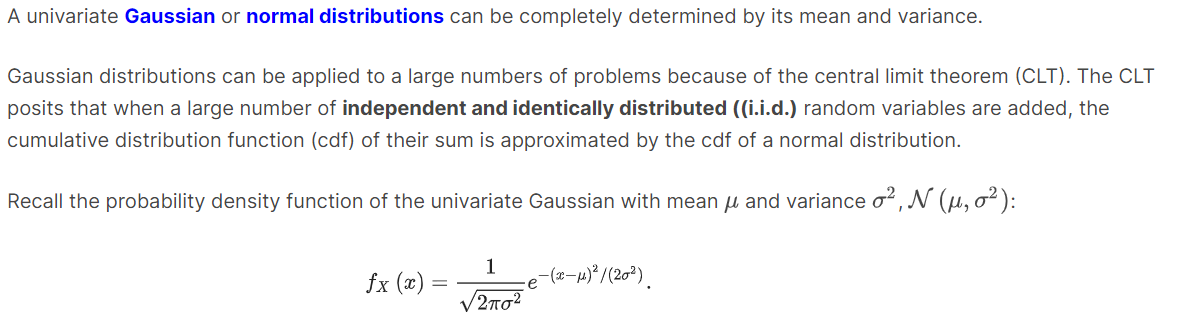
  



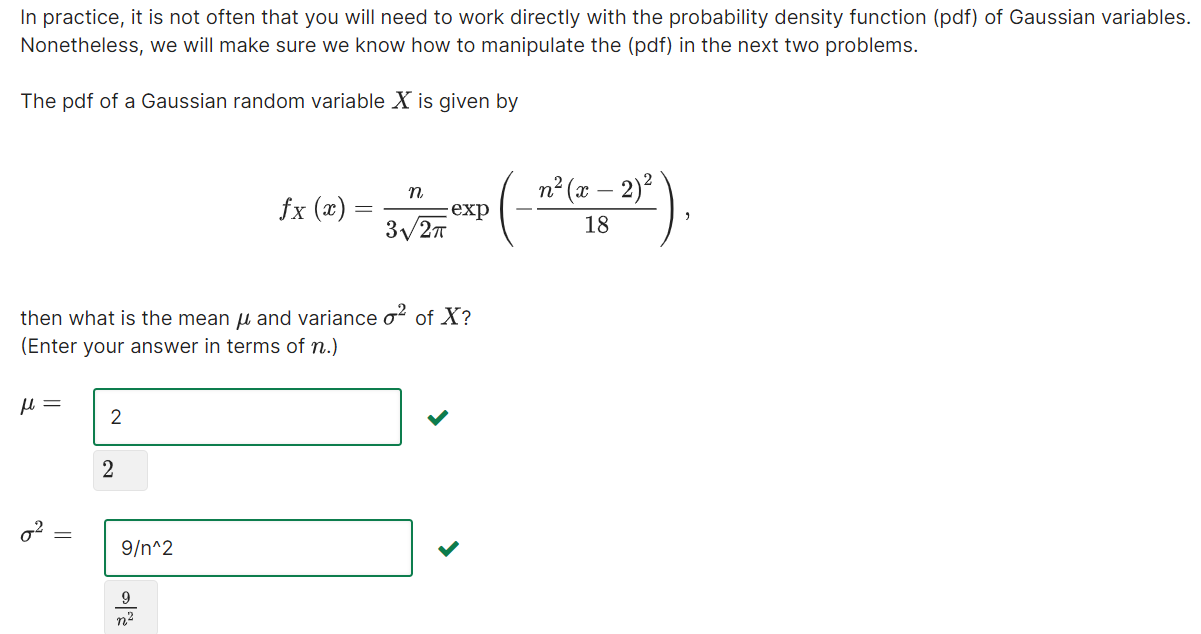


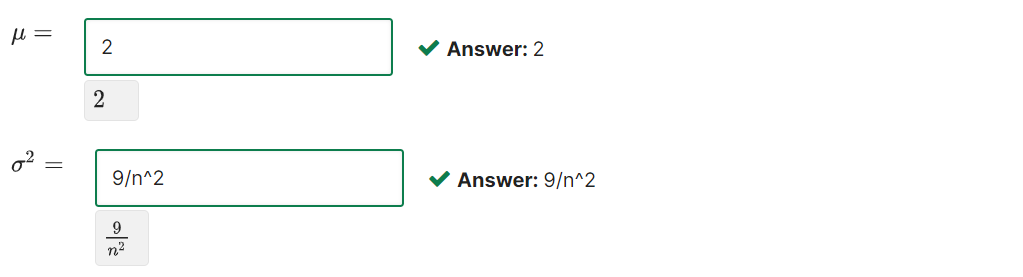
### Probability: Univariate Gaussians

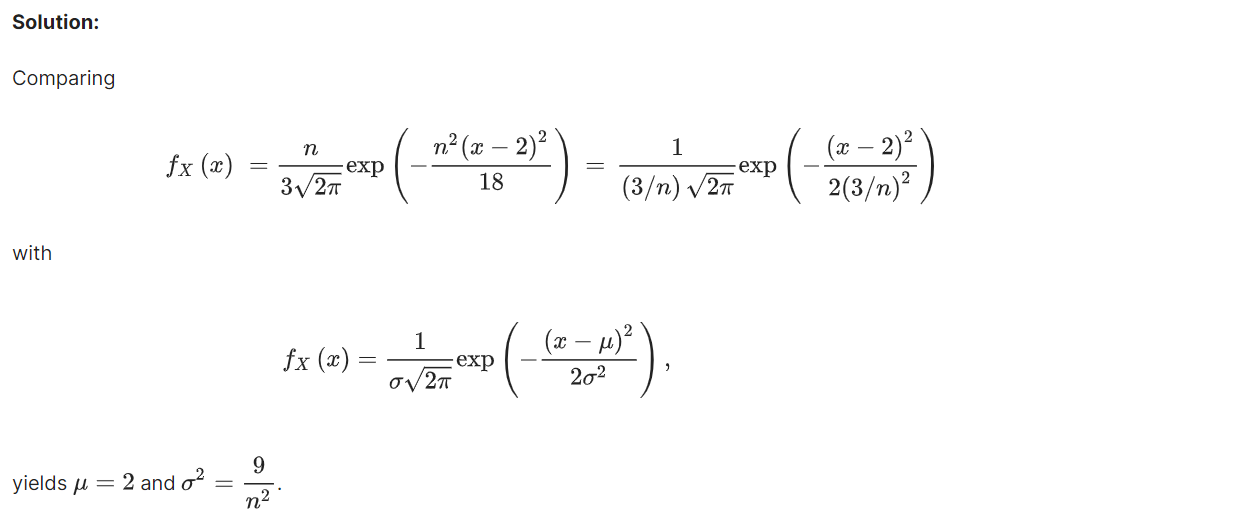
#### Univariate Gaussians



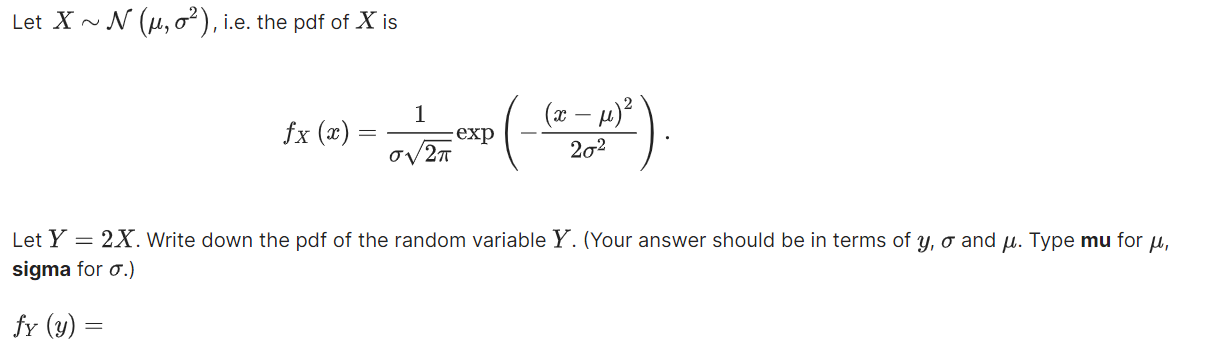
Question 1:

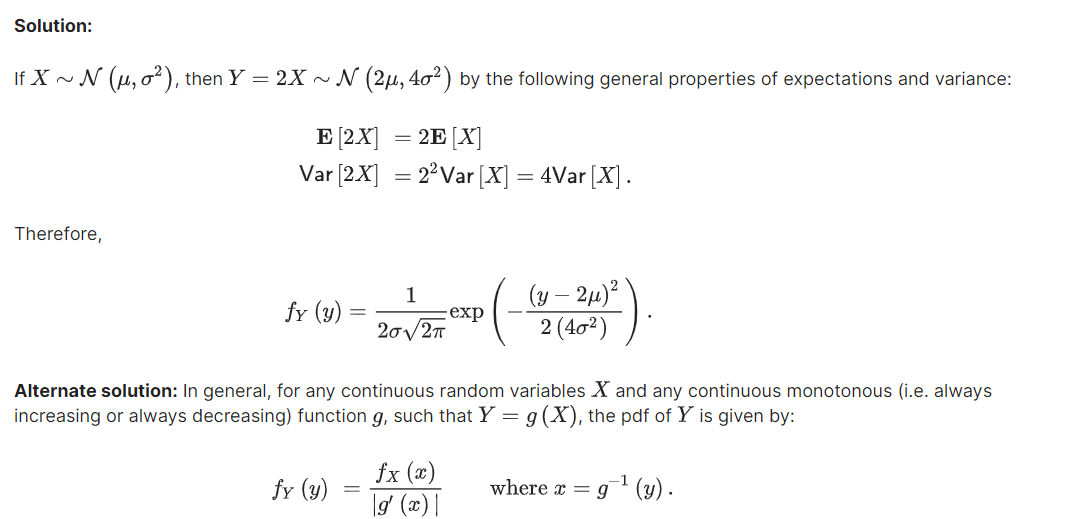


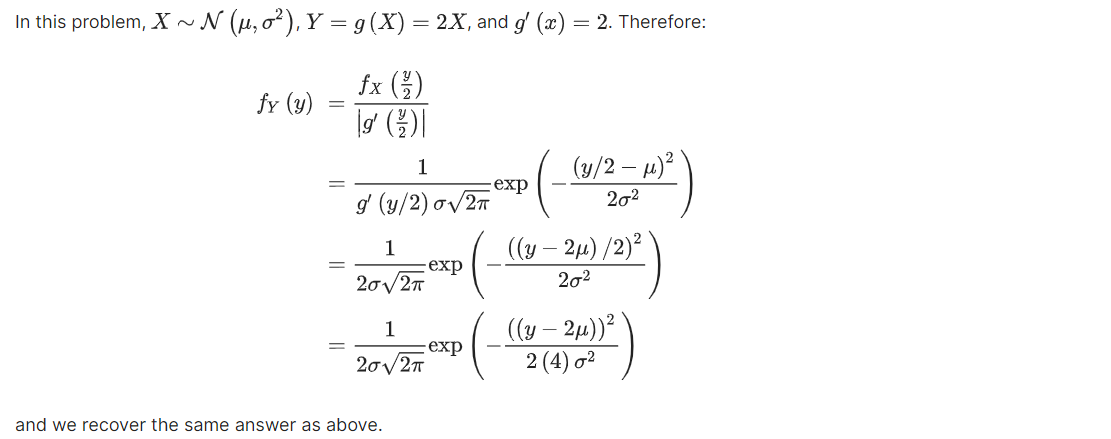




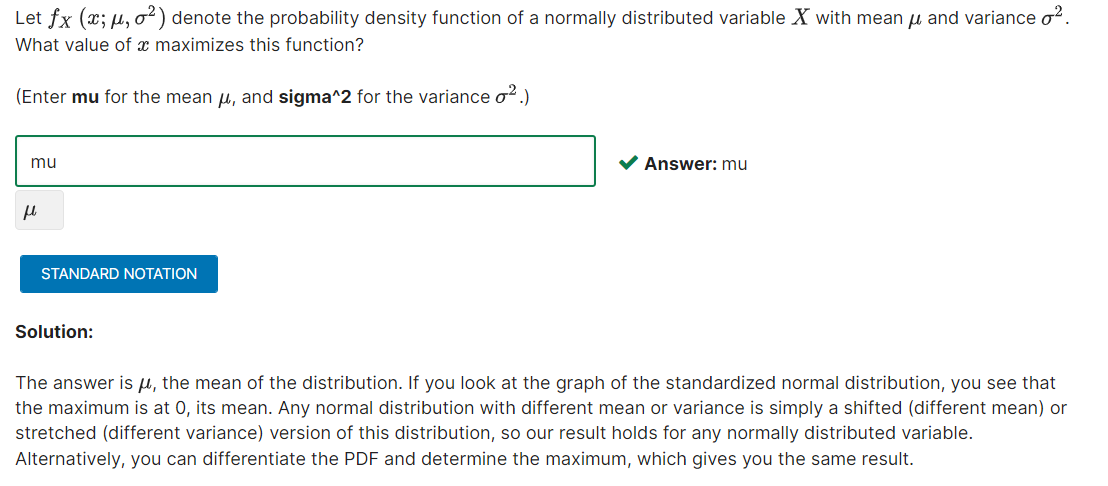
Question 2:



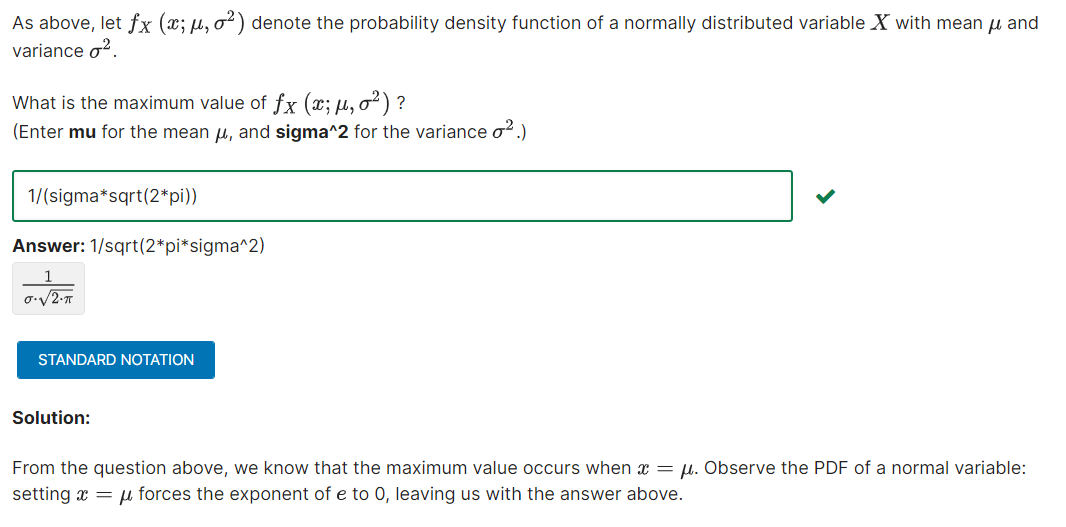




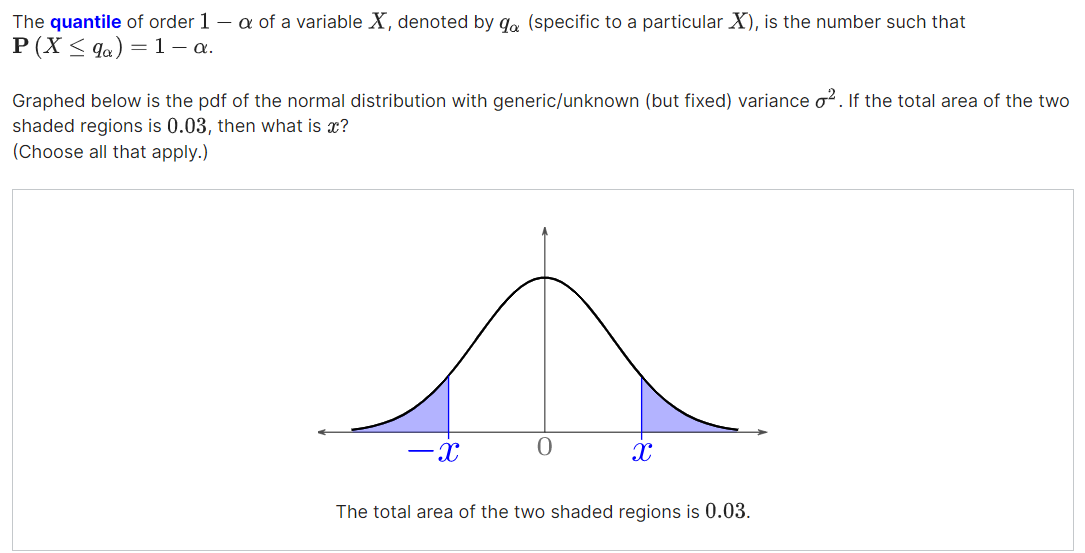
#### Argmax

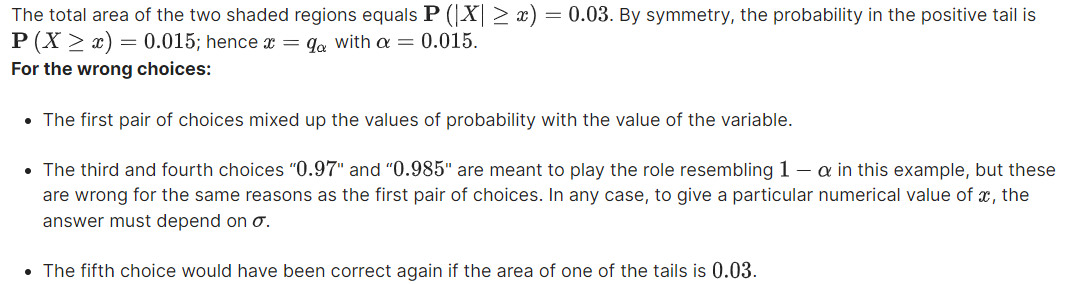


#### Maximum of pdf

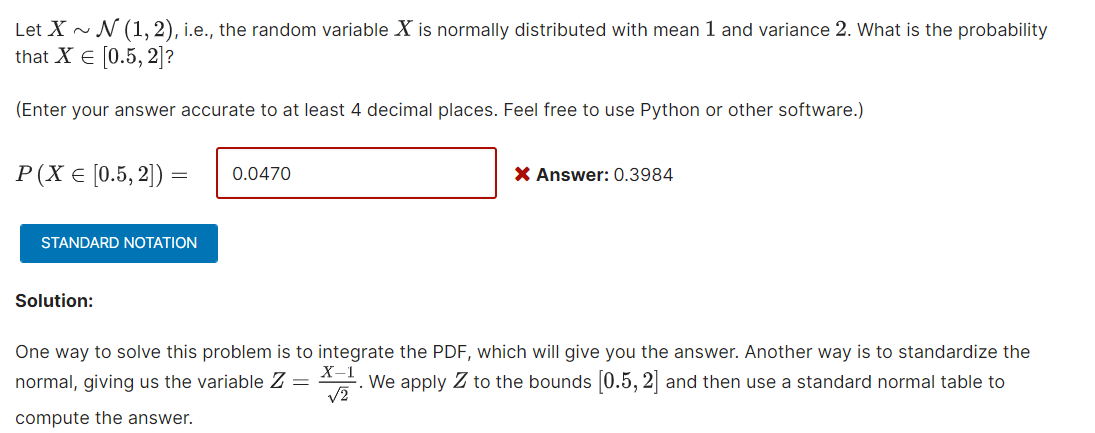


#### Quantiles



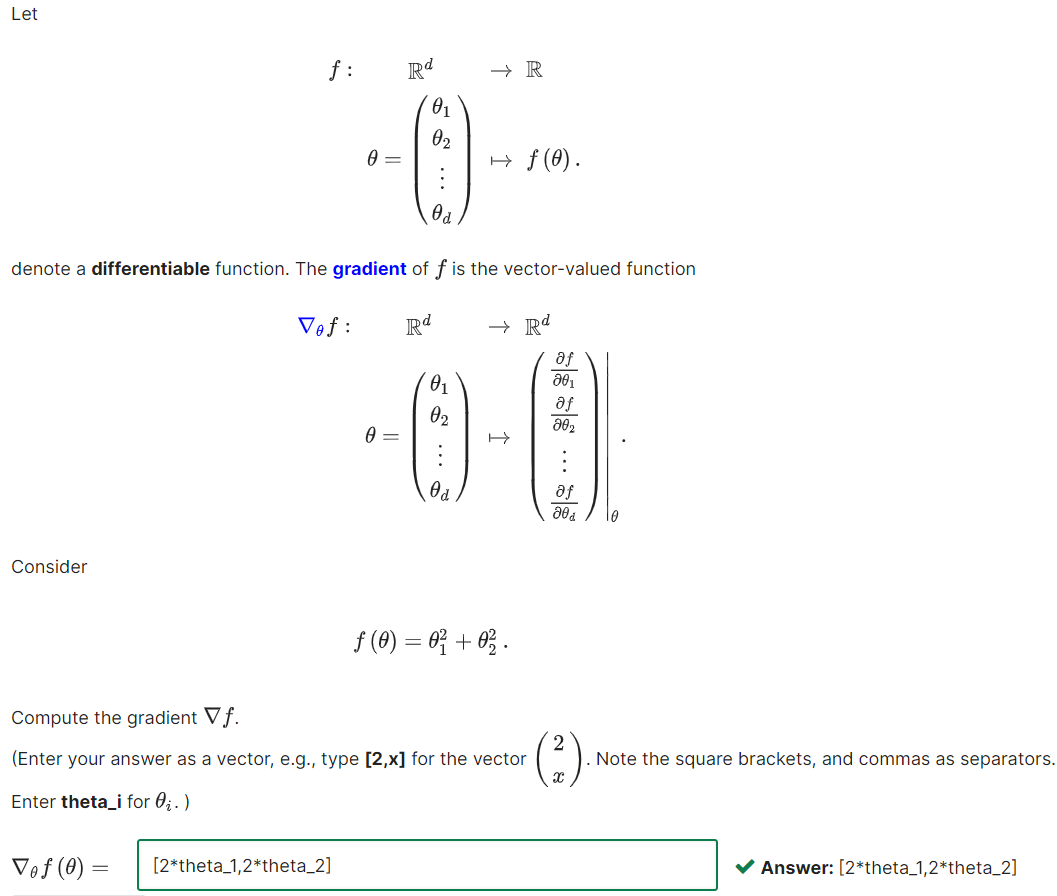
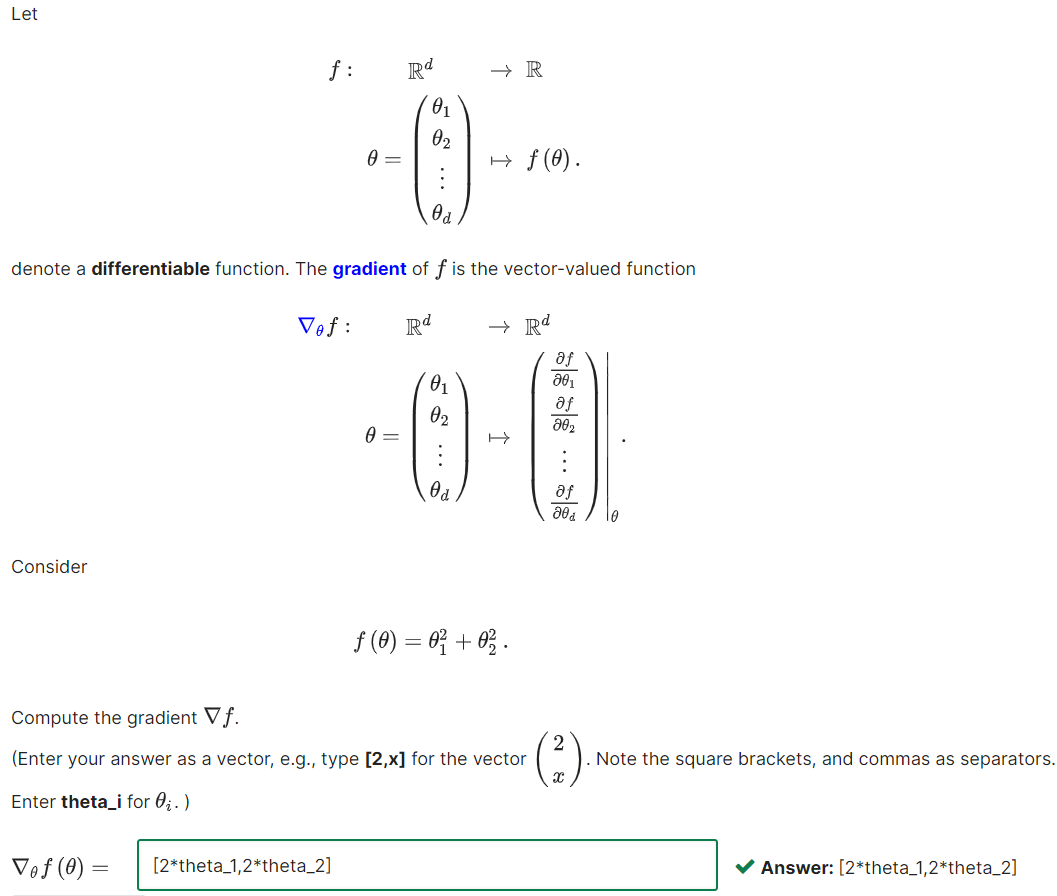


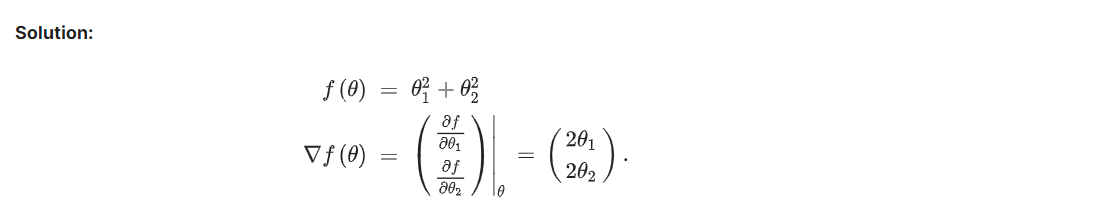
#### Probability



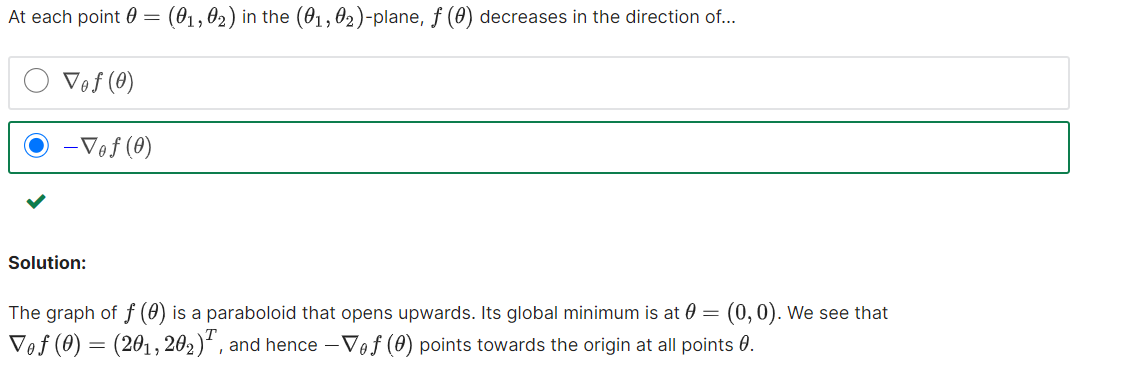
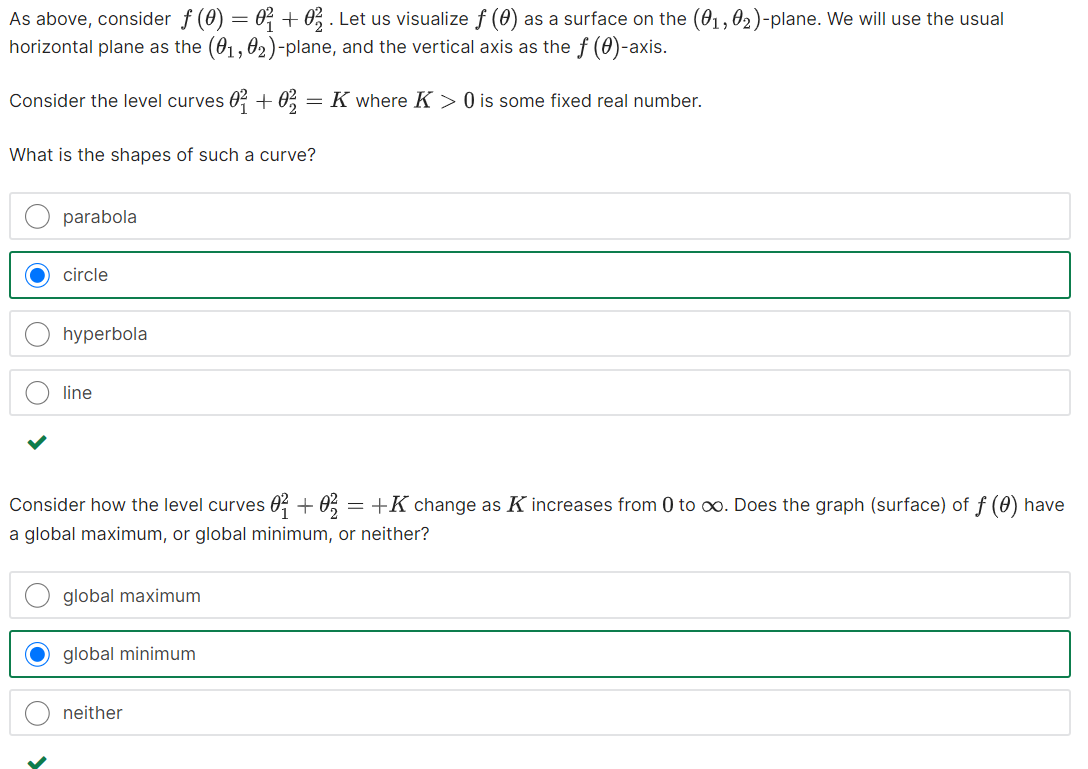
### Gradients and Optimization – Độ dốc (đạo hàm) và Tối ưu

#### Multivariable Calculus Review: Simple Gradient

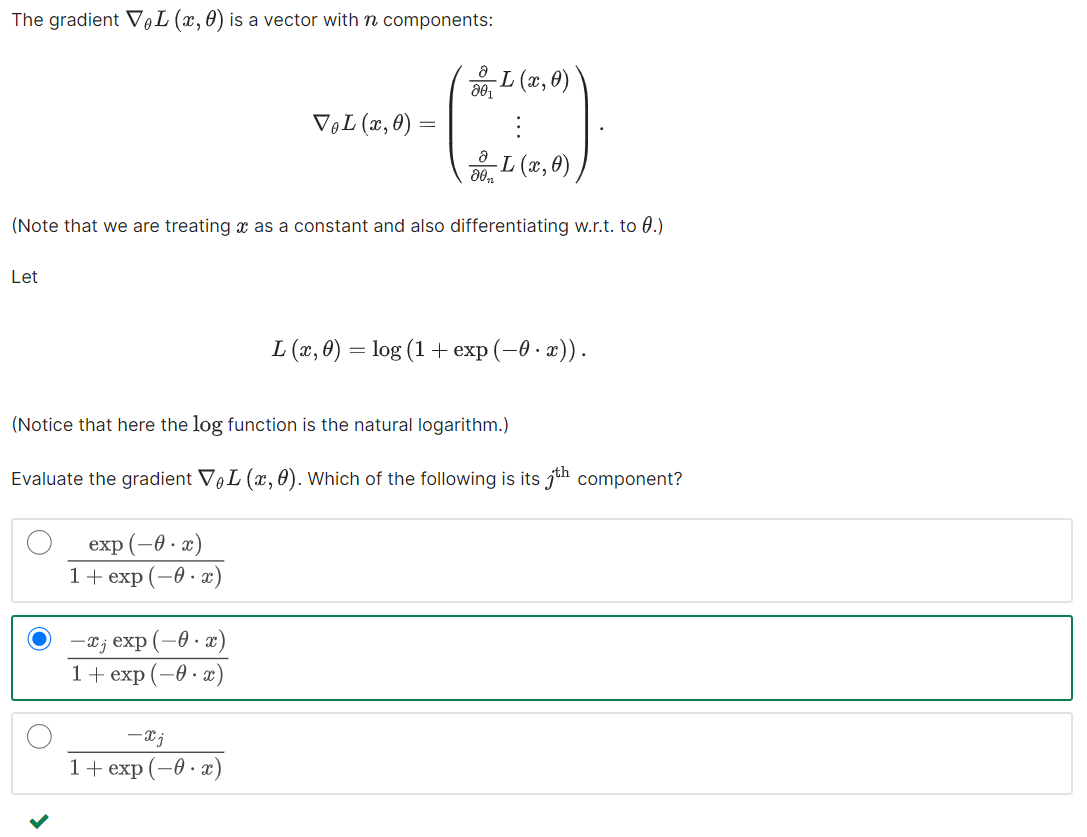




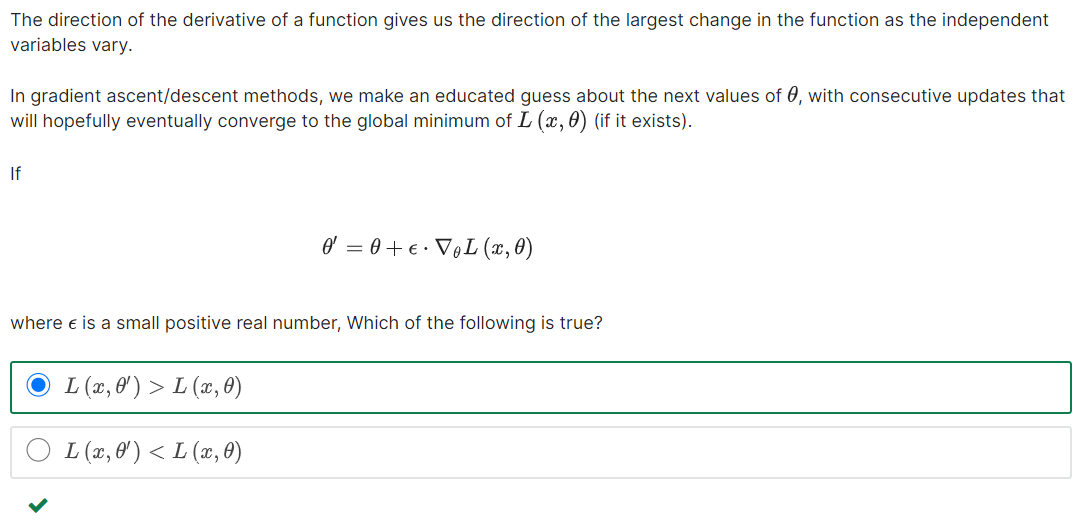
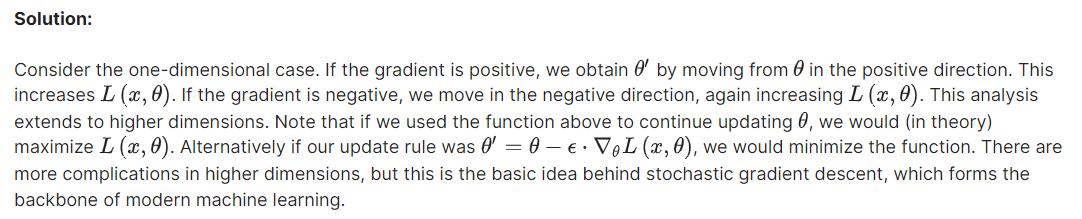
#### Geometric Picture of the Function



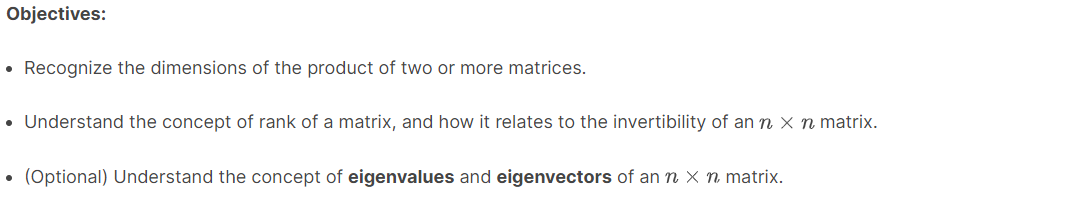
#### Compute the Gradient



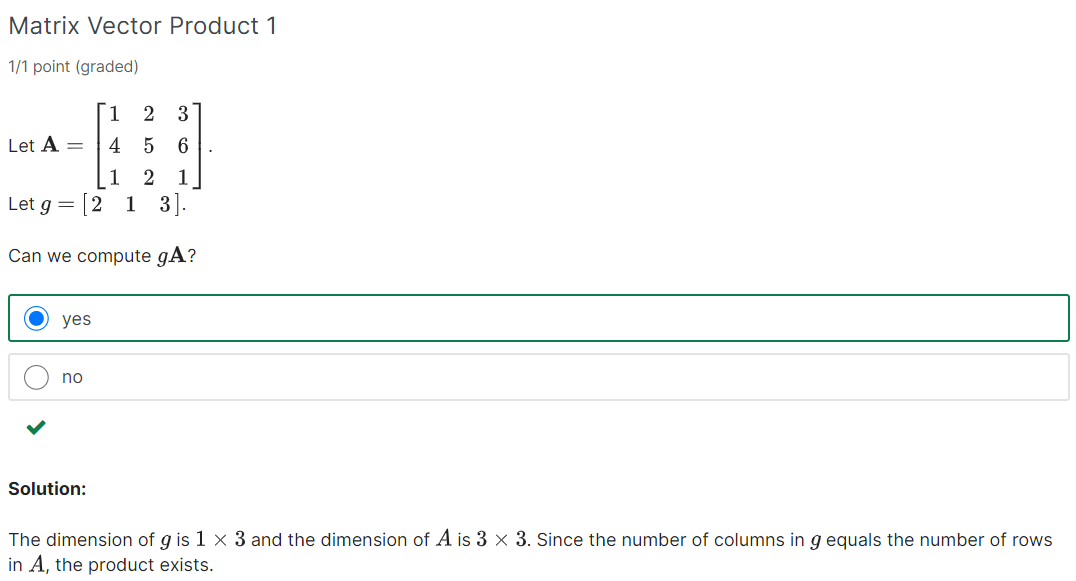
#### Gradient Ascent or Descent – Điểm cực đại, cực tiểu

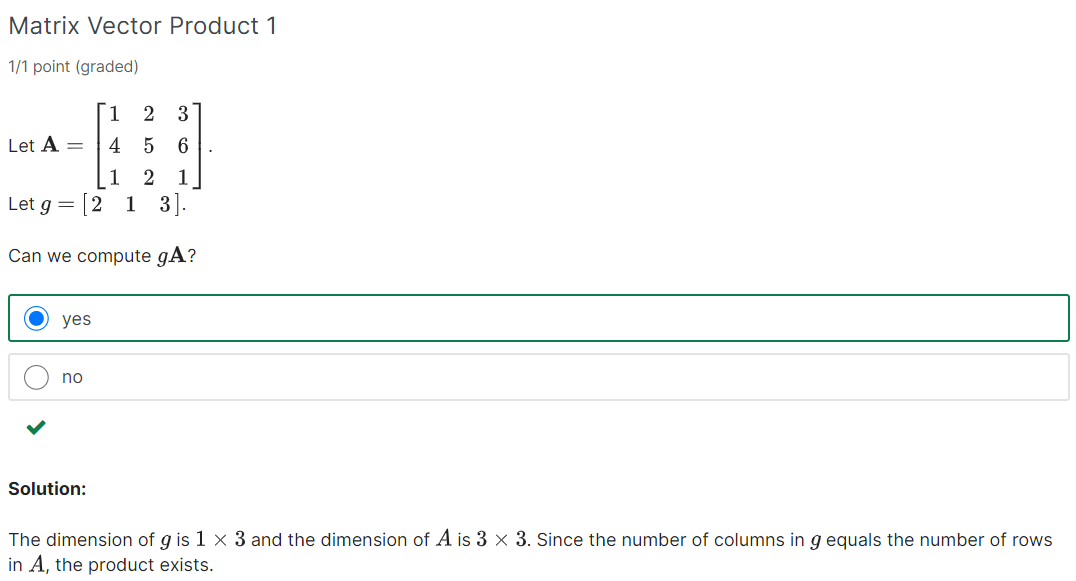
 

### Matrices and Vectors

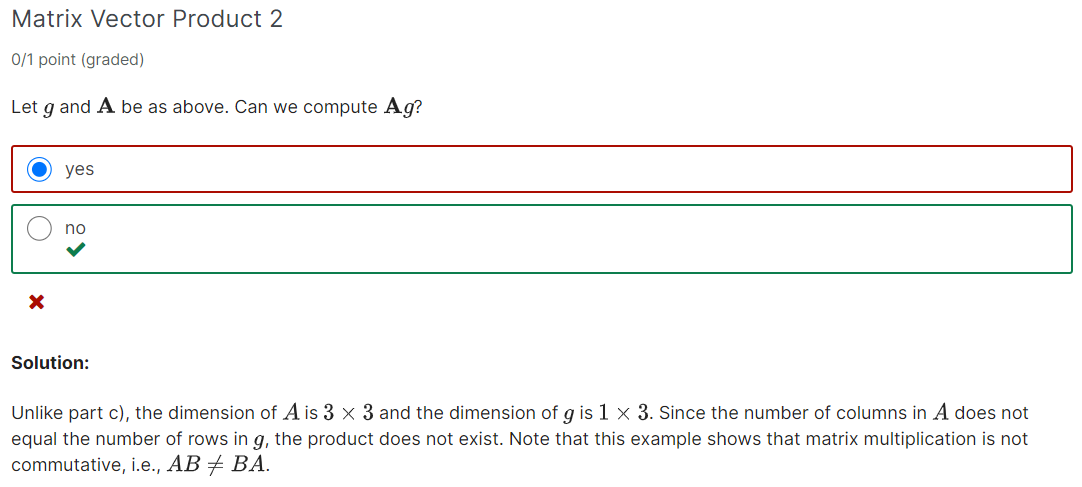


Question 1:

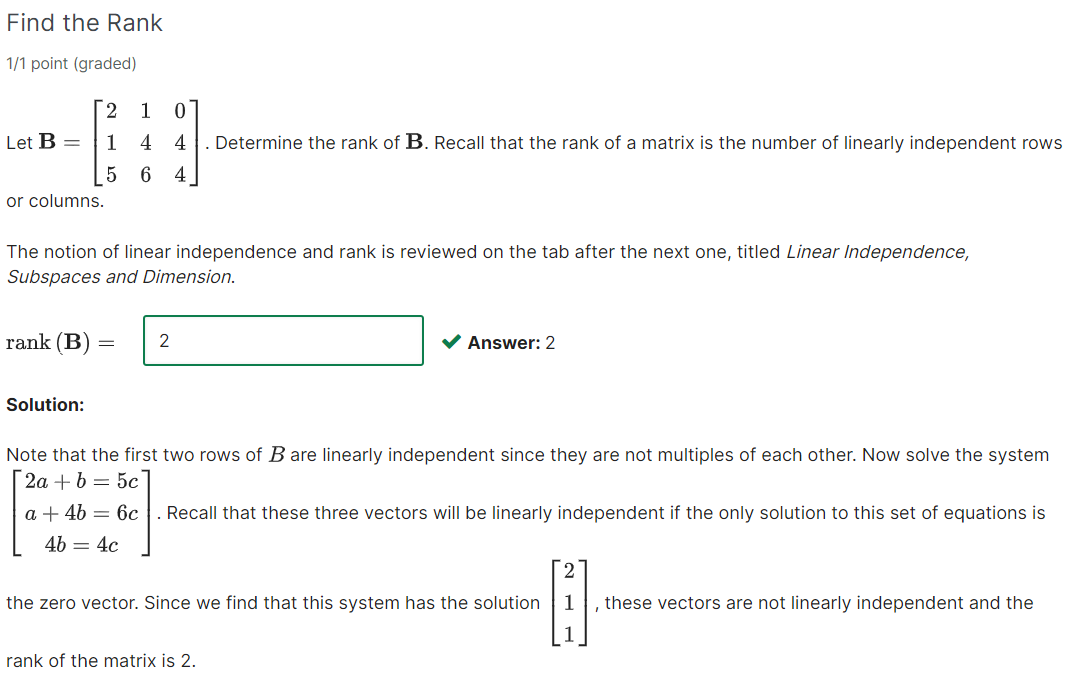




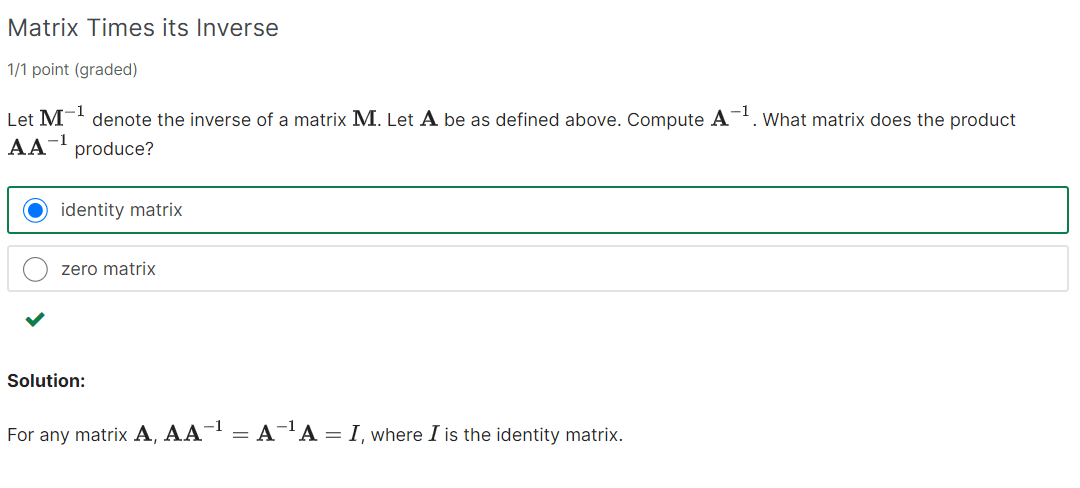
Question 2:

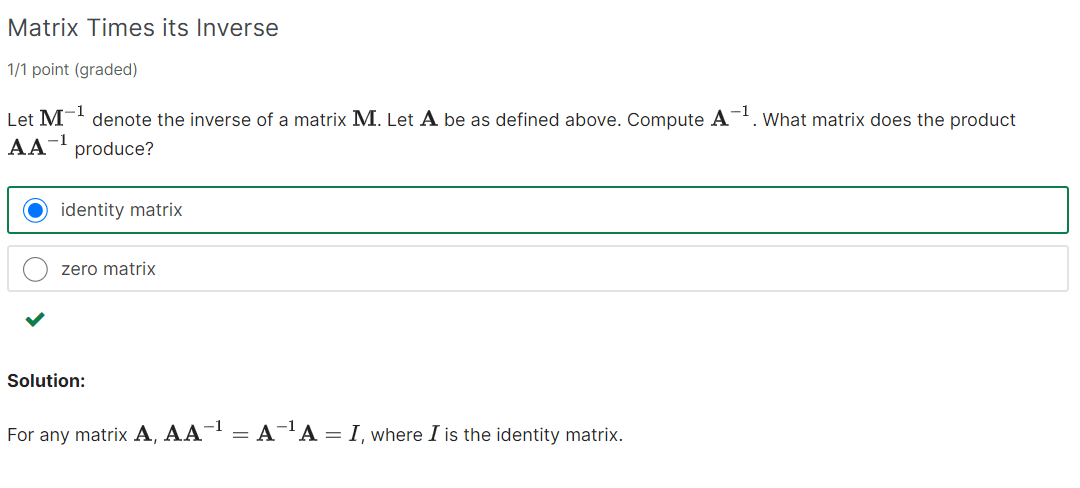


#### Find the rank



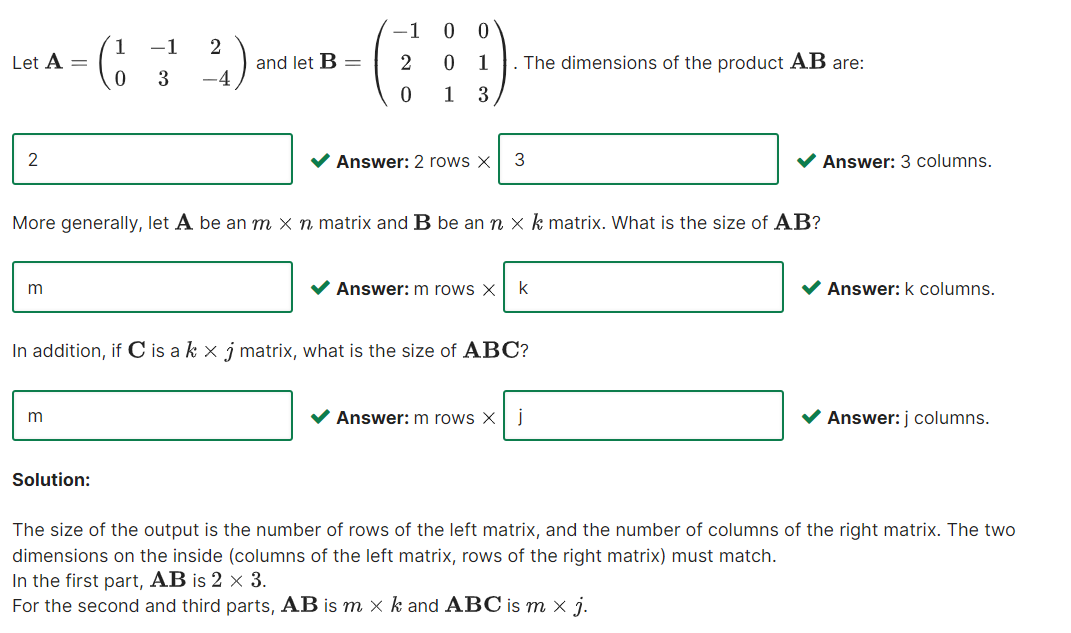
#### Matrix Times that inverse



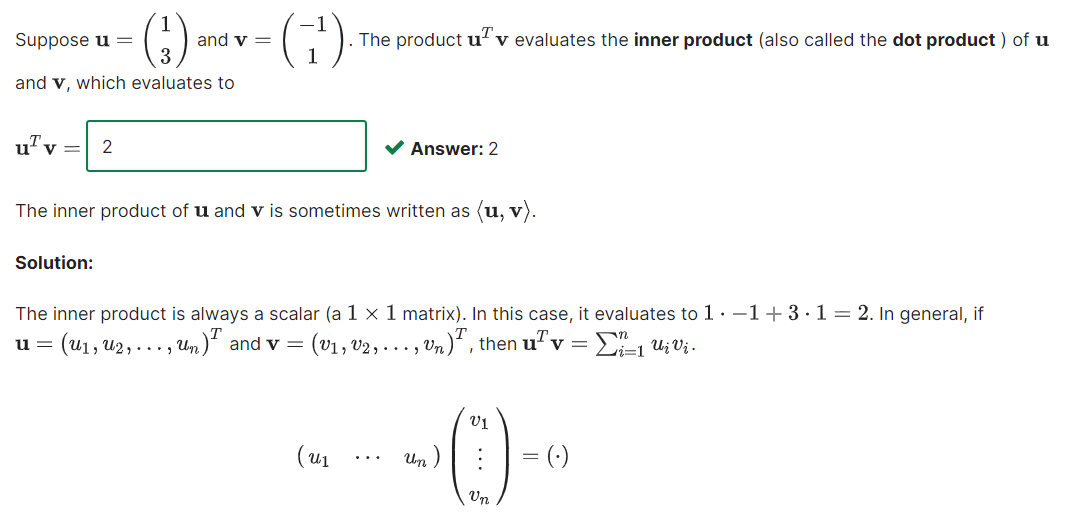


### Matrix Multiplication

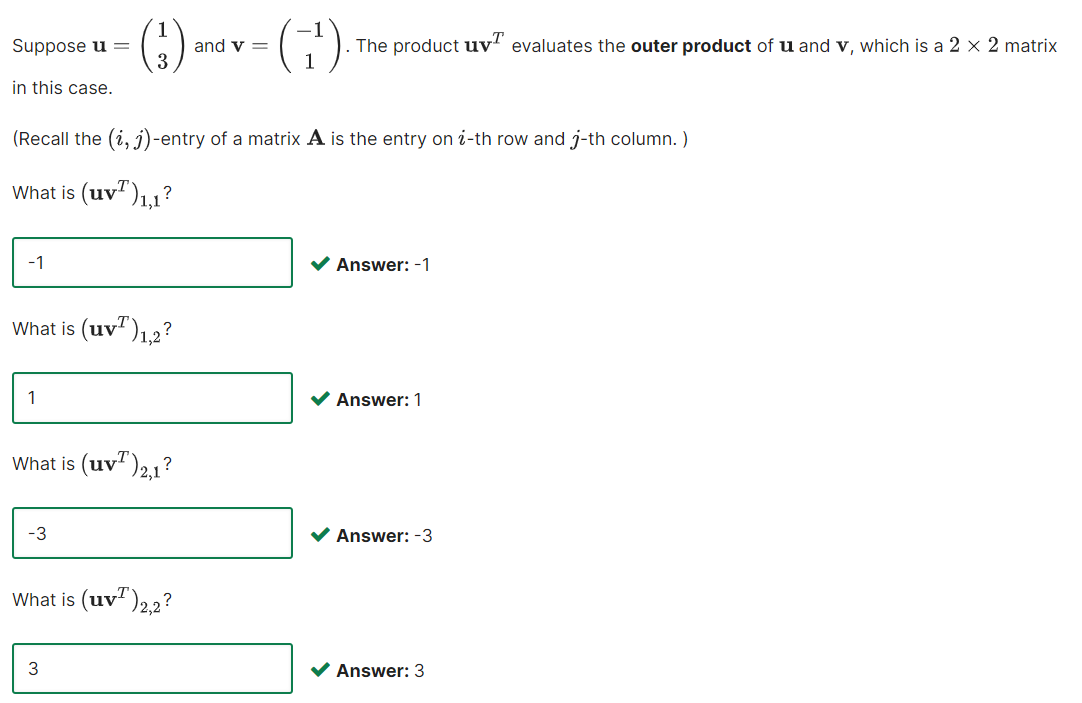
#### Matrix Multiplication

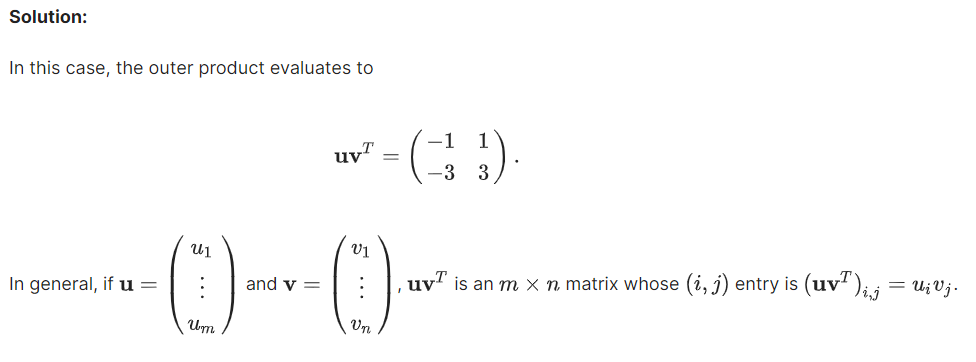


#### Vector Inner product

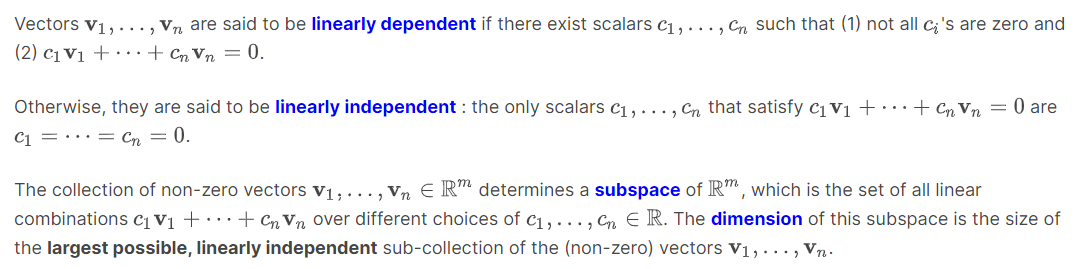


#### Vector Outer product

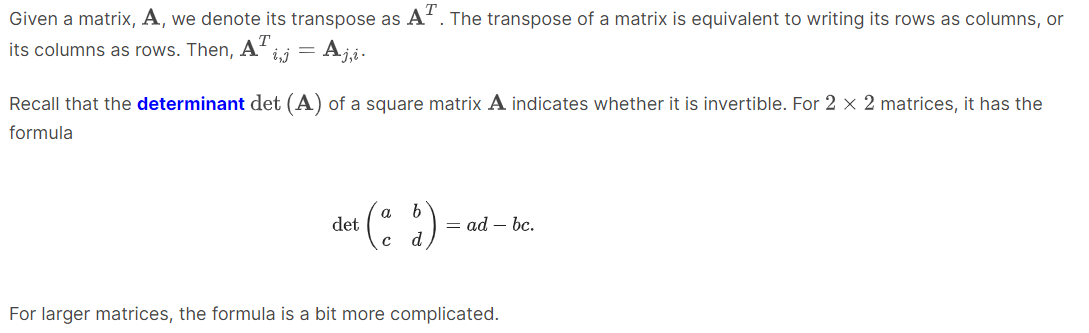




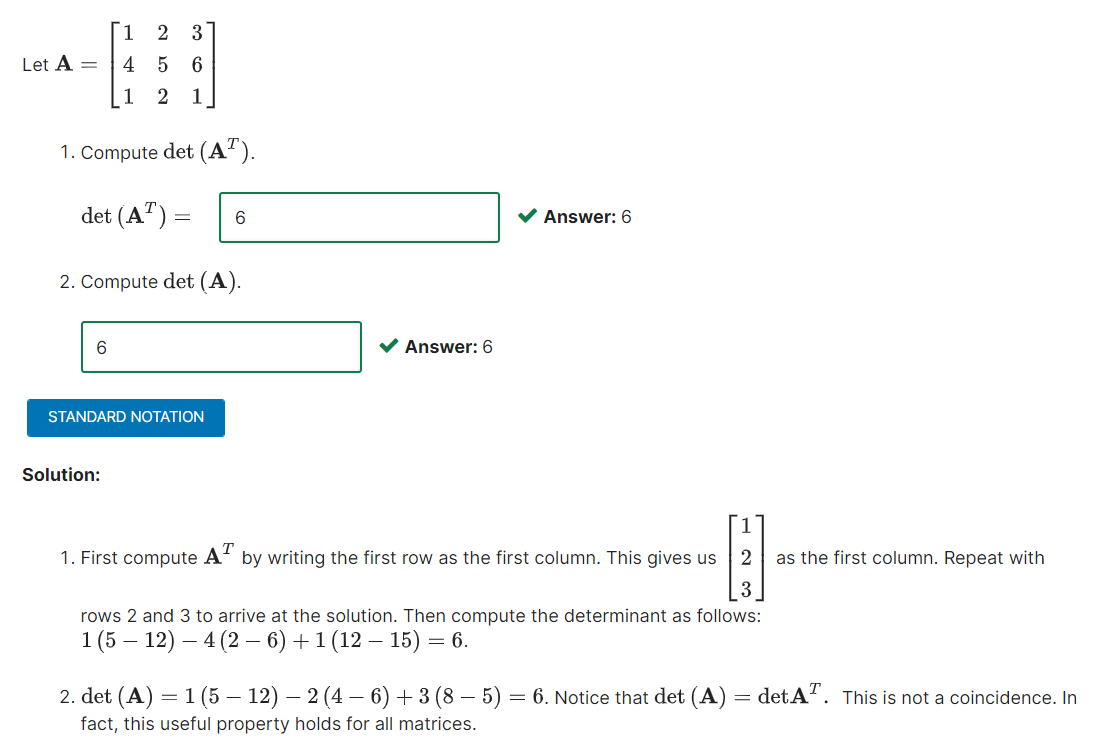
### Linear Independence, Subspaces and Dimension



### Determinant

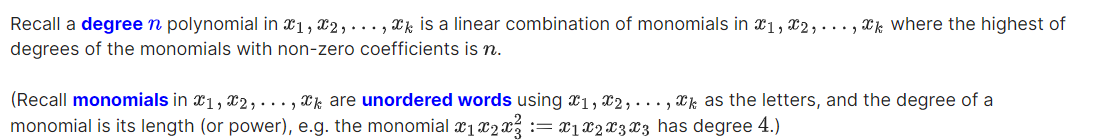
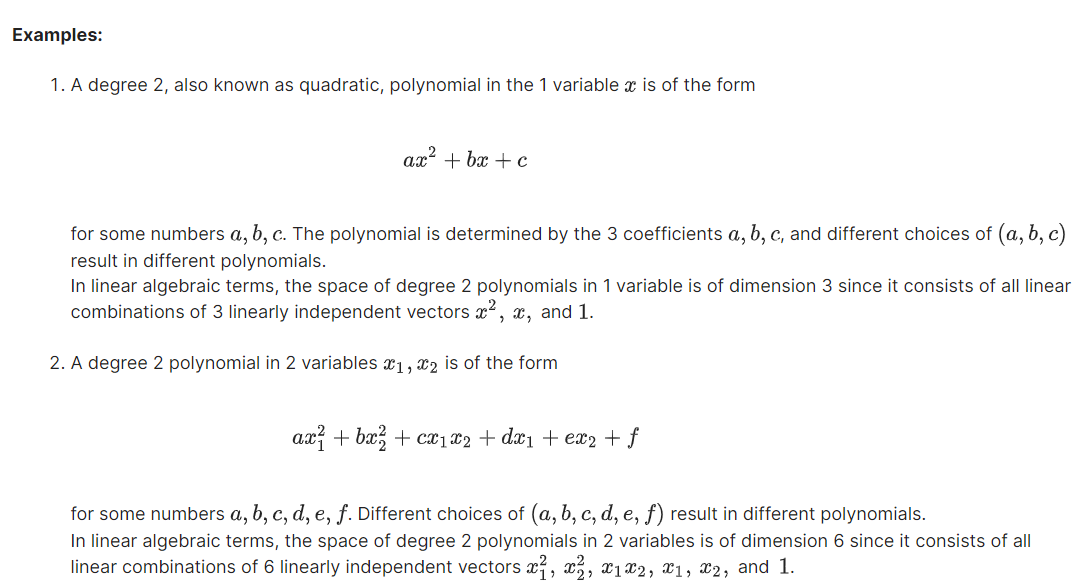


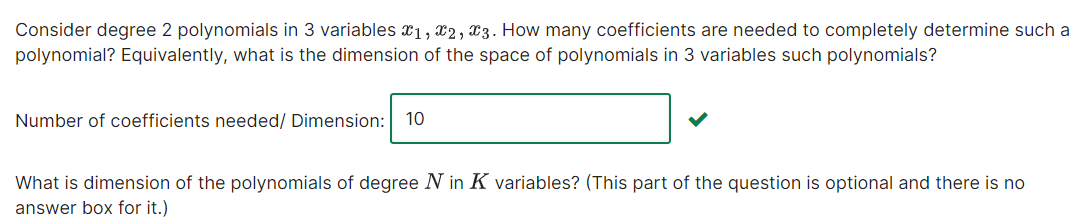
#### Compute the Determinant



### Interlude: Polynomials (hàm số căn bậc) and Geometric

#### Quadratic Polynomials

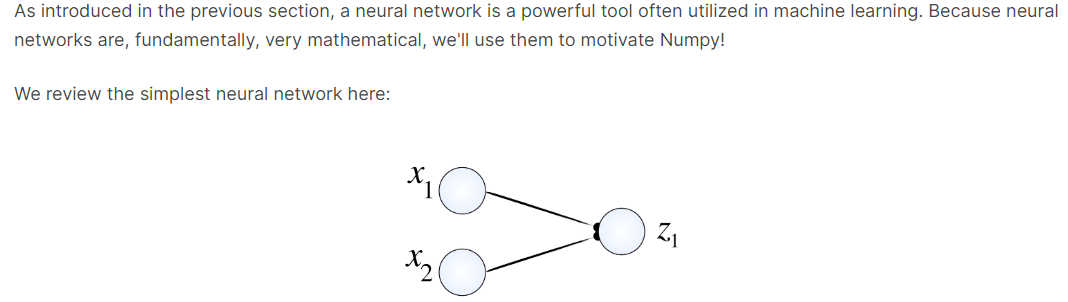


## Project 0

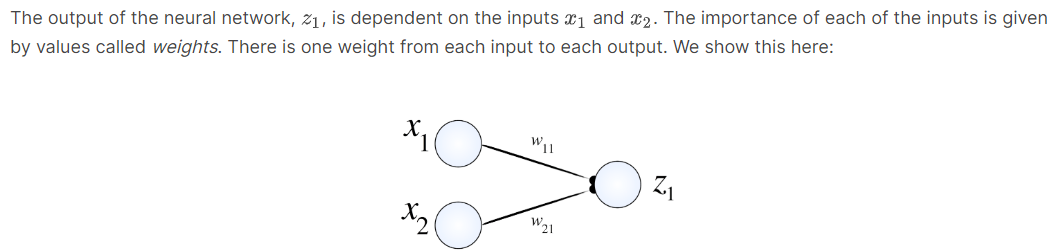
#### Goals of the project

The goal of this project 0 is to help you set up your python environment and to give you an introduction to the mechanics of the online grading system.

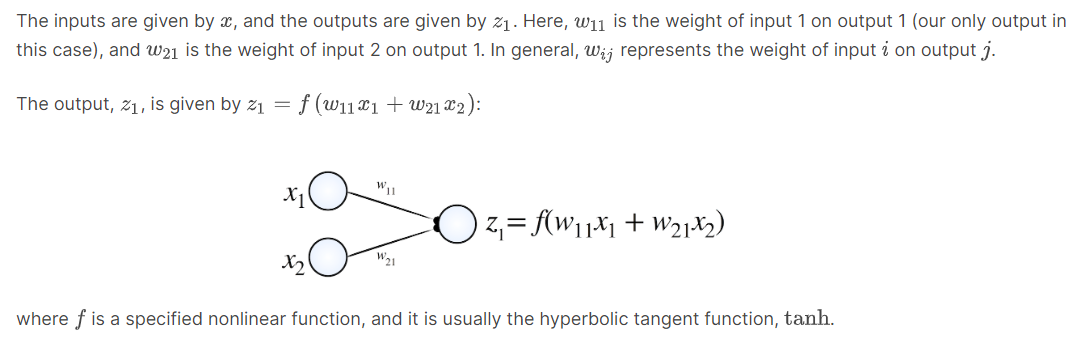
#### Initial requirements

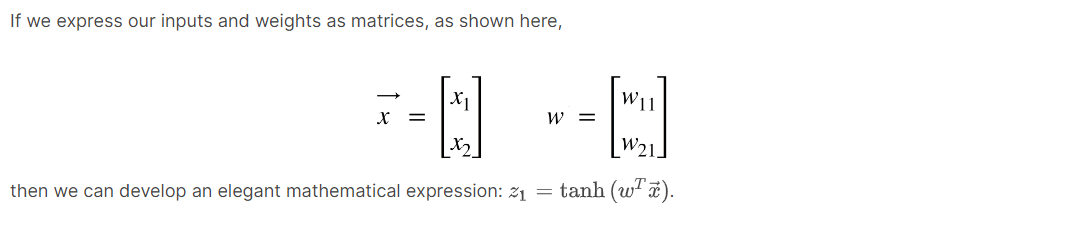


The output of the neural network, Z1 is dependent on the inputs x1 and x2. The importance of each of the inputs is given by values called *weights*. There is one weights from each input to each output. We show this here:



The inputs are given by x, and the outputs are given by z1. Here, w11 is the weight of input 1 on output 1 (our only output in this case), and w21 is the weight of input 2 on out 1. In general, wij, represents the weight of input i on output j



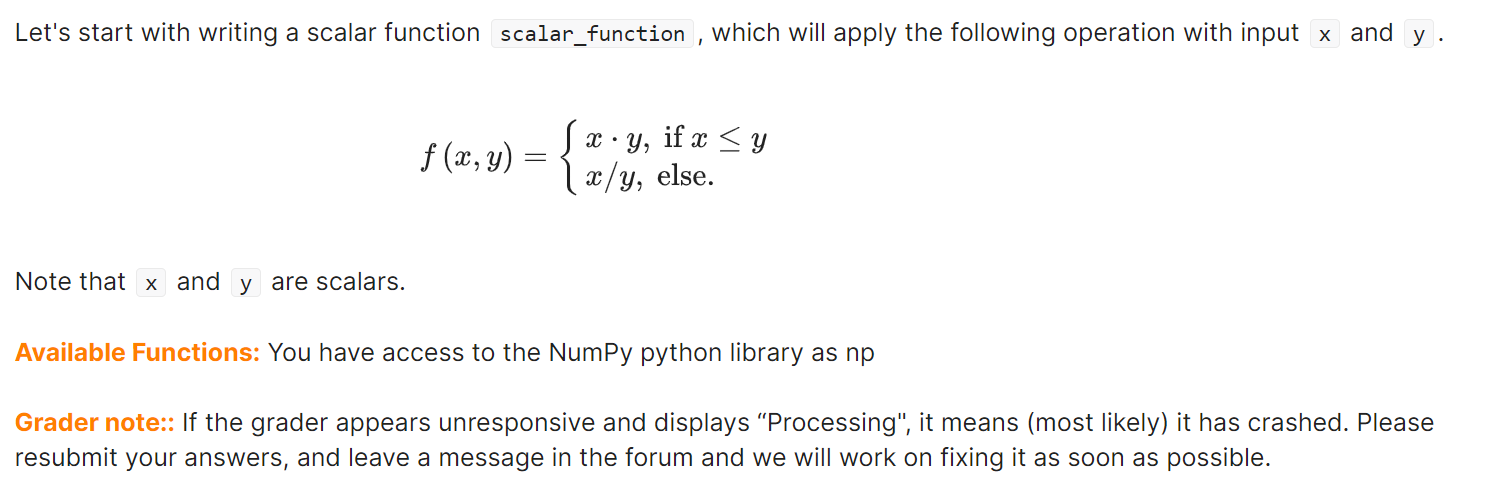


### 5. Neural Network

|  |
| --- |
| *Convolutional Neural Network* |

### 6. Vectorize function

#### Scalar function



#### Vector function



# Unit 0 Unit 1 Linear Classifiers and Generalizations

## Homework 1. Introduction to Machine Lear

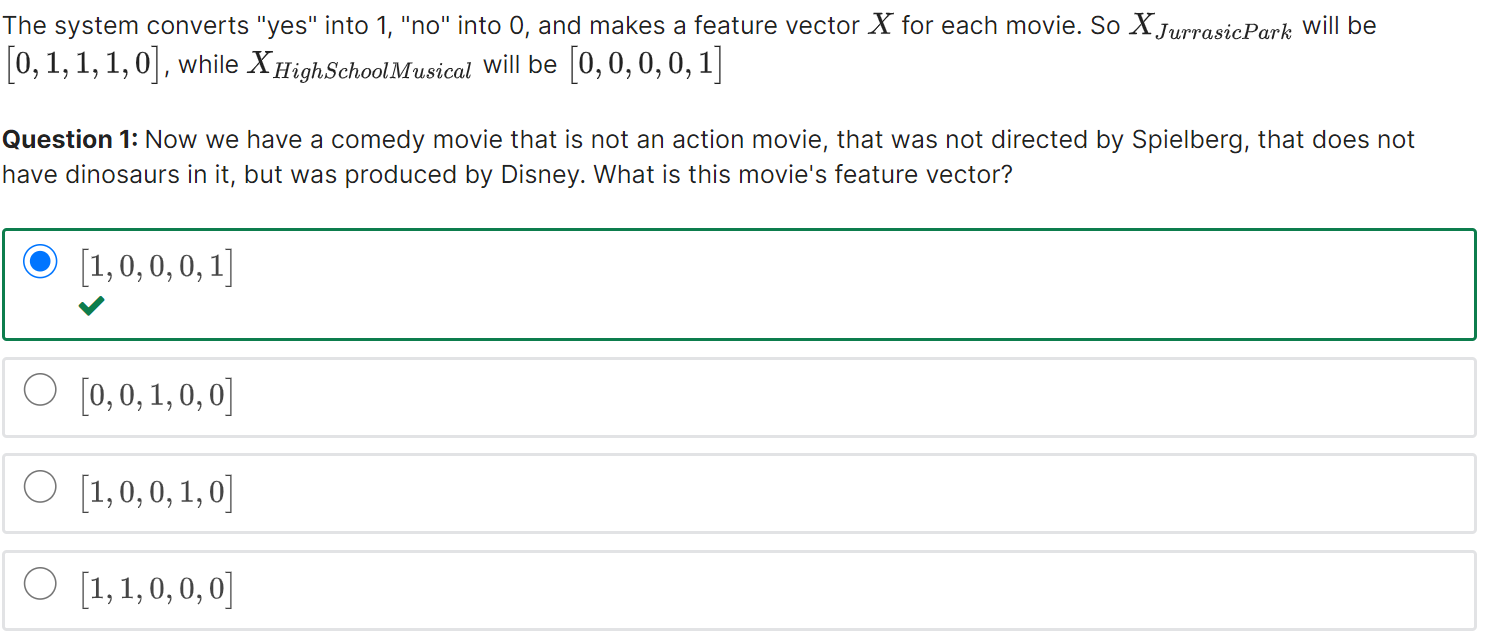
### 5. A Concrete Example of a Supervised Learning Task

#### Feature Vector Demystified 1

We have a movie recommending system that reads description of each movie and determines some important characteristics of the movie. In particular, it examines whether each of the criterion below is true for that movie:

* Is it a comedy movie?
* Is it an action movie?
* Was the movie directed by Spielberg?
* Do dinosaurs appear in the movie?
* Is it a Disney film?

For example, when the recommending system reads descriptions of "Jurassic Park", the answers for the five questions above will be "no, yes, yes, yes, no." On the other hand if the recommending system reads descriptions of "High School Musical", the answers will be "no, no, no, no, yes"



#### Feature Vector Demystified 2

Question 2:What is the dimension of the feature vector of this movie?

**Solution:** Each feature vector has length 5, so its dimension is 5.

#### Training Set vs Test Set 1

Question 1:What is the label of movie 1, based on the fact that John likes the movie?

**Solution:** If John likes the movie, the label is 1. Otherwise, it is -1.

**Question 2:**What movies are in the **training set**? Select all those apply.

**Solution:** Movies whose labels are available are in the training set. Thus movies 1,2,3,4 are in the training set.

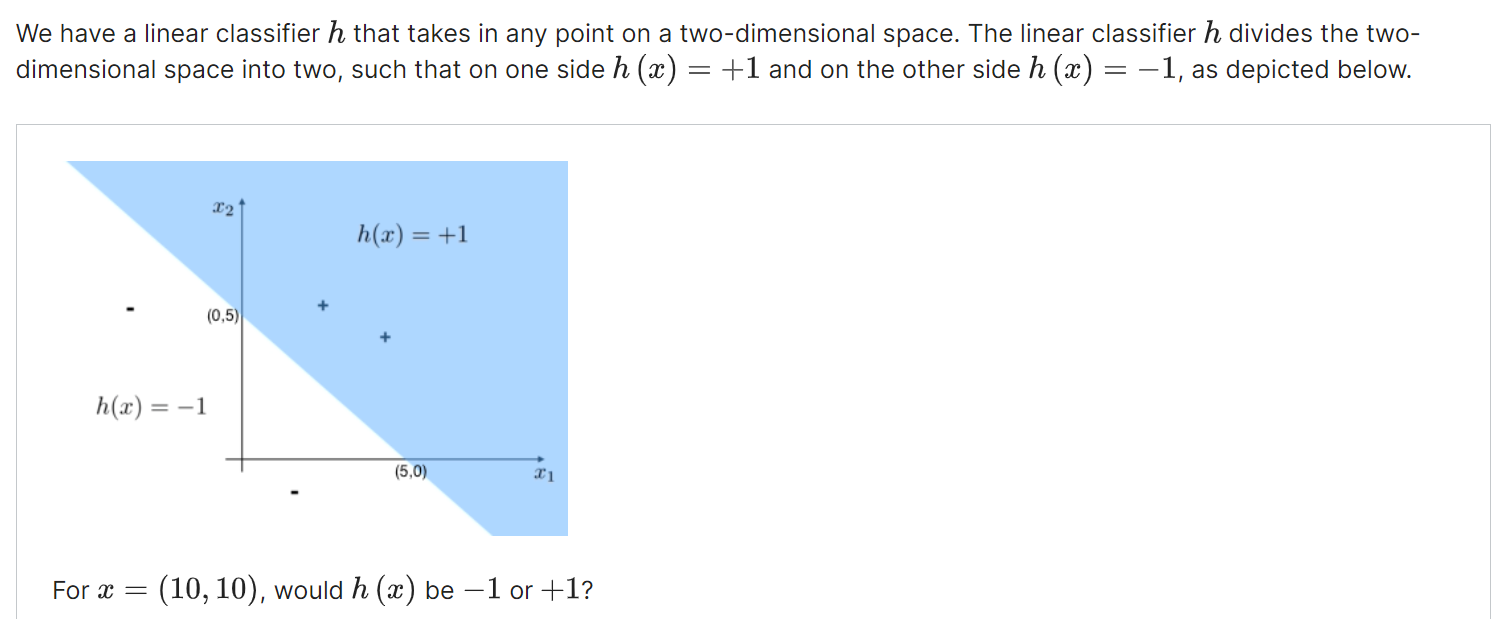
**Question 3:**What movies are in the **test set**? Select all those apply.

Movies whose labels are not yet available are in the test set. Thus movies 5,6,7 are in the test set. Remember that it is our end goal to predict these movies' labels.

### 6. Introduction to Classifiers: Let's bring in some geometry!



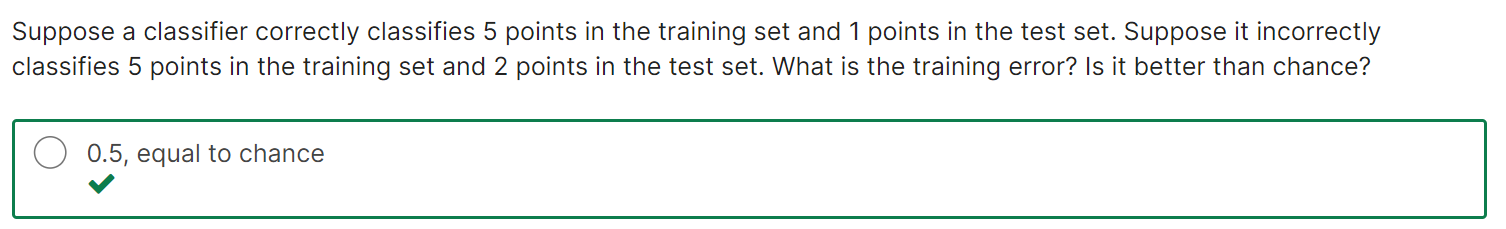
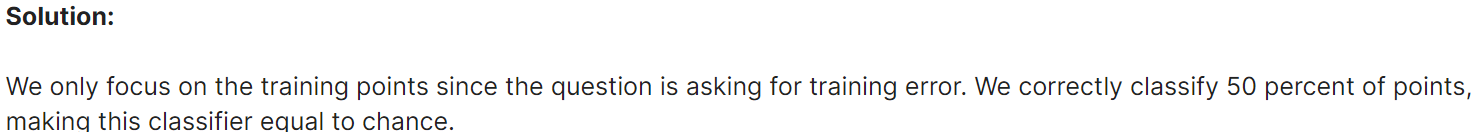
#### Linear Classifier



As an aside, classifiers need not be linear. They can be of any shape!

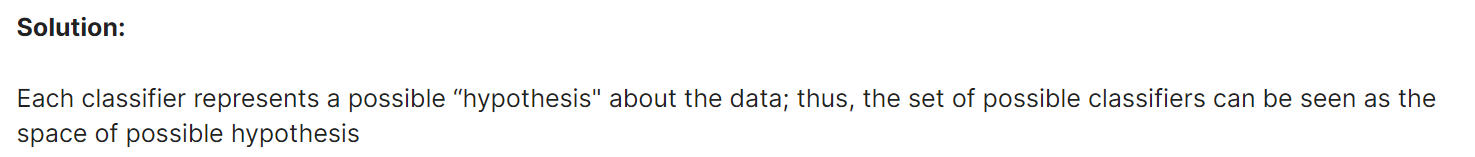
**Solution: +1**

#### Training Error

#### Hypothesis Space

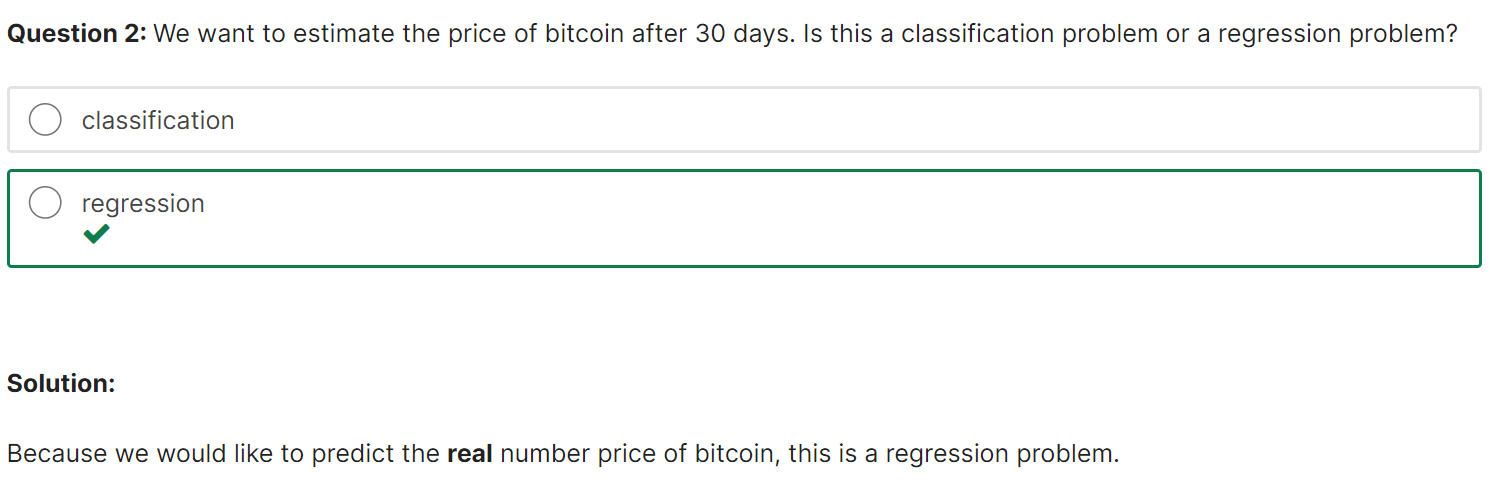




### 7. Different Kinds of Supervised Learning: classification vs regression

**Classification** maps **feature vectors** to **categories**. The number of categories need not be two - they can be as many as needed. **Regression** maps feature vectors to **real numbers**. There are other kinds of supervised learning as well.

#### Classification or Regression?

#### Different Types of Learning

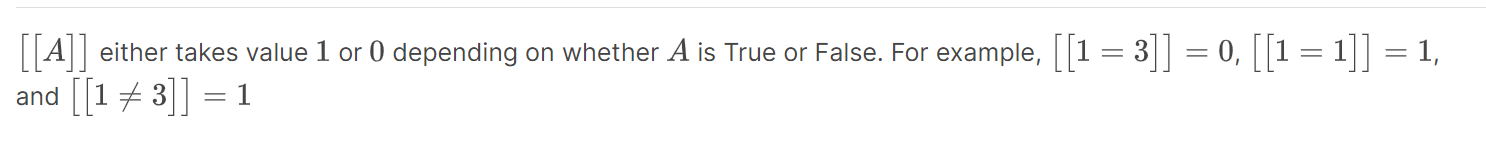
1. Labelled training and test examples -> supervised learning
2. Using knowledge from one task to solve another task -> Tranfer learning
3. Learning to navigate a robot -> reinforcement learning
4. Deciding which examples are needed to learn -> Active learning
5. Data with no annotation -> unsupervised learning
6. Training and test examples with limited annotation -> semi-supervised learning

* Fully labelled training and test examples corresponds to **supervised learning**.
* Limited annotation is **semi-supervised learning**, and no annotation is **unsupervised learning.**
* Using knowledge from one task on another task means you're “**transferring" information**.
* Learning how to navigate a robot means learning to act and optimize your actions, or **reinforcement learning**.
* Deciding which examples are needed to learn is the definition of **active learning**.

## Homework 2. Linear Classifiers and Perceptron Algorithm

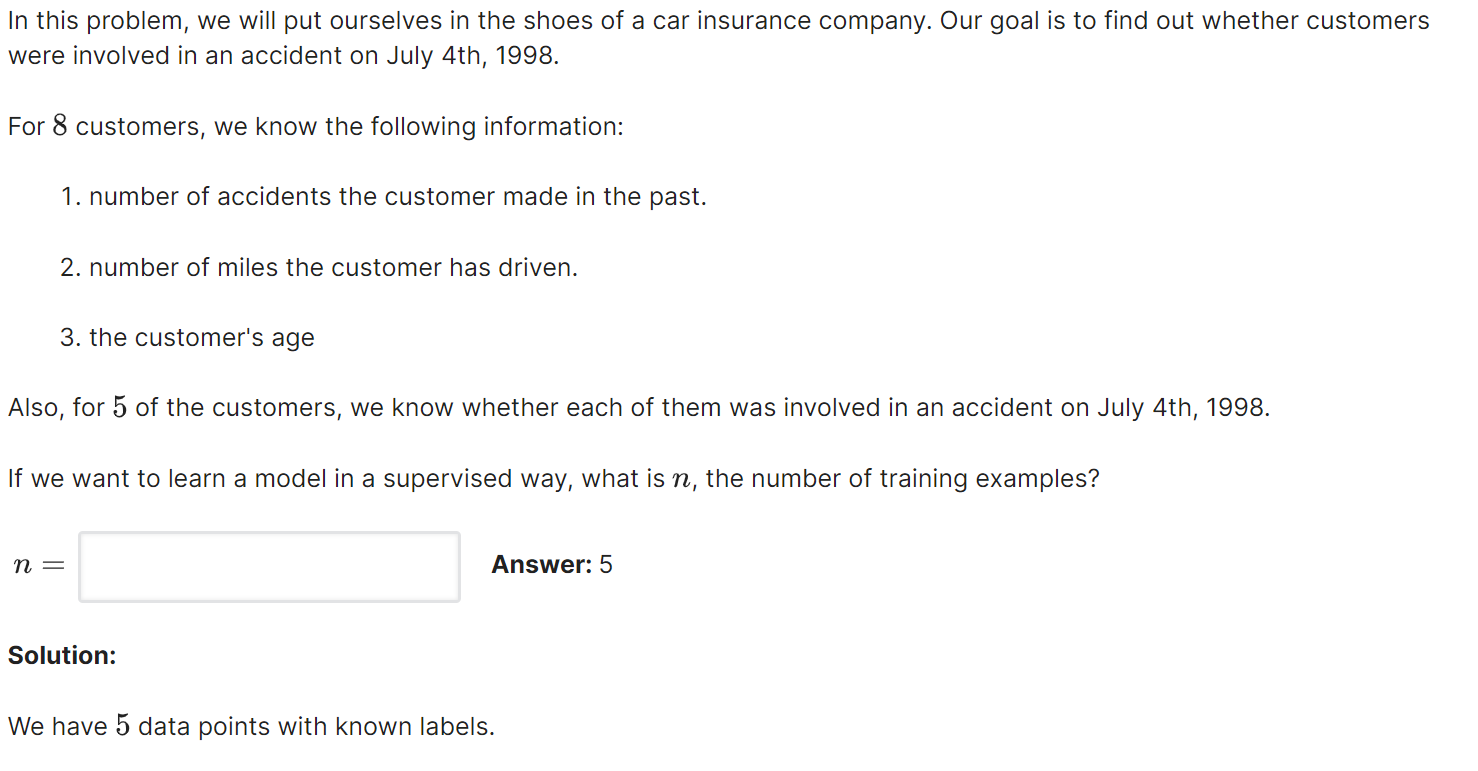
At the end of this lecture, you will be able to

* understand the concepts of Feature vectors and labels, Training set and Test set, Classifier, Training error, Test error, and the Set of classifiers
* derive the mathematical presentation of linear classifiers
* understand the intuitive and formal definition of linear separation
* use the perceptron algorithm with and without offset

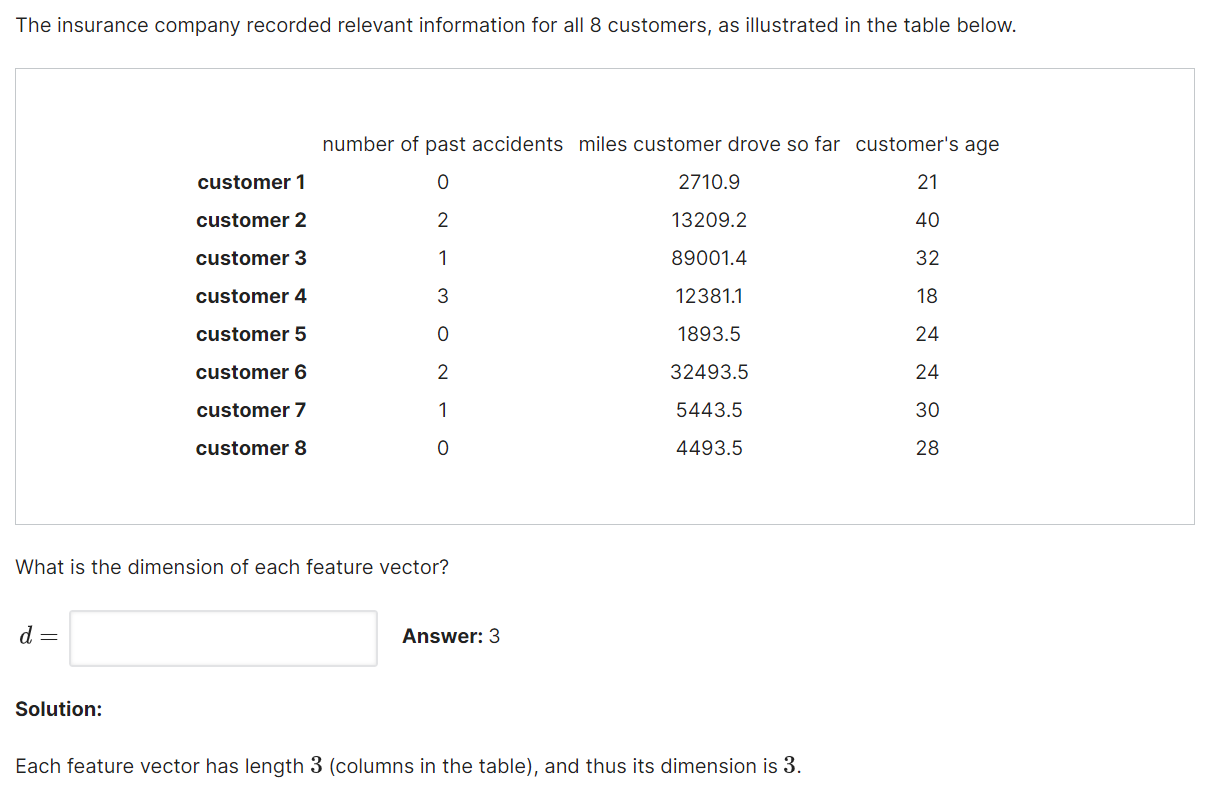


#### Concept Review Problem: car accident prediction

**Question 1:**

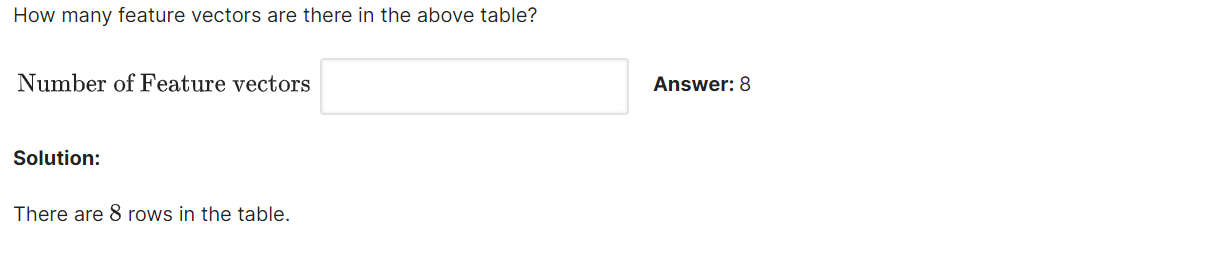


**Question 2:**



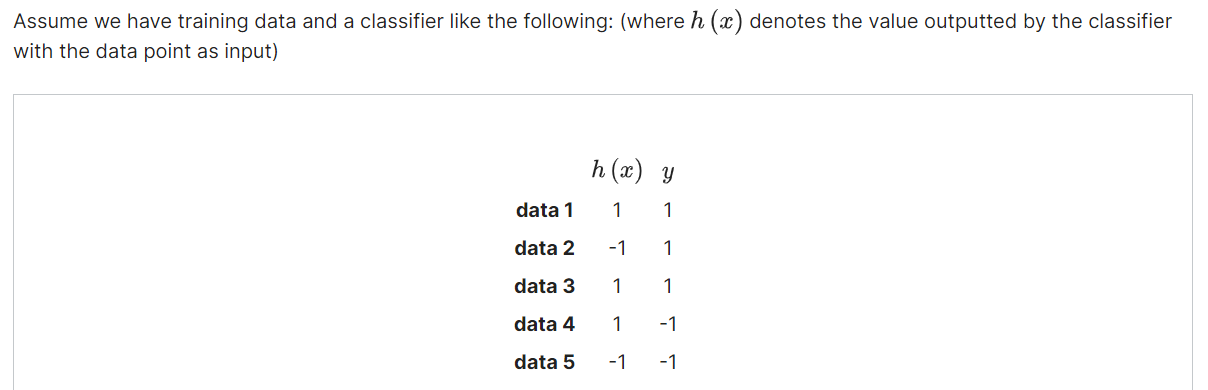
**Solution:** Each feature vector has length 3 (columns in the table), and thus its dimension is 3.

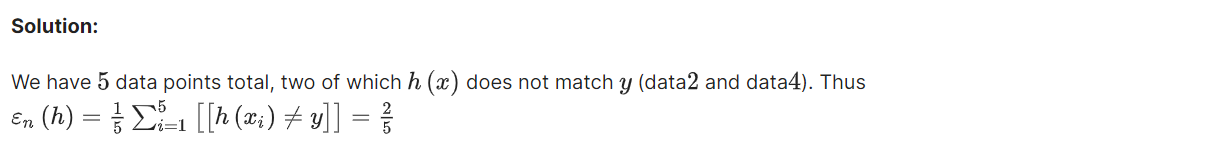
**Question 3:**



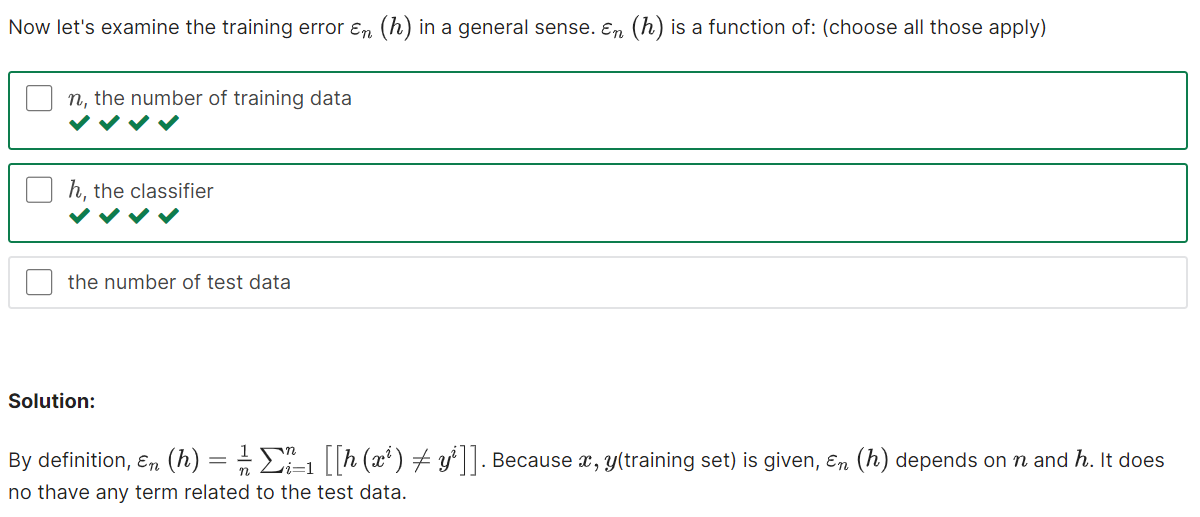
#### Concept Review Problem: Classifier and Training Error

**Question 1:**



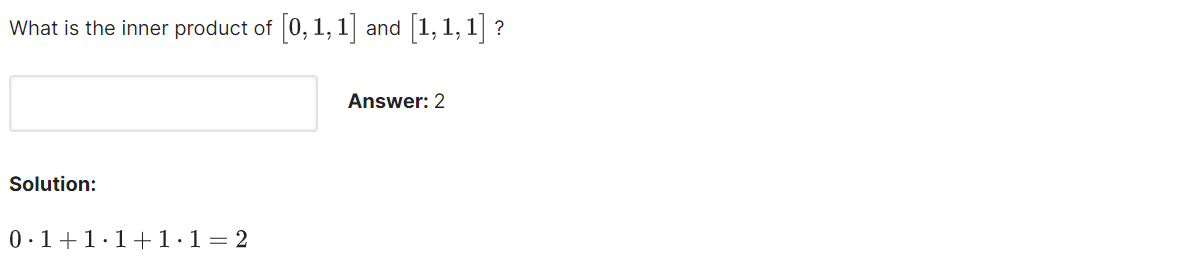


**Question 2:**

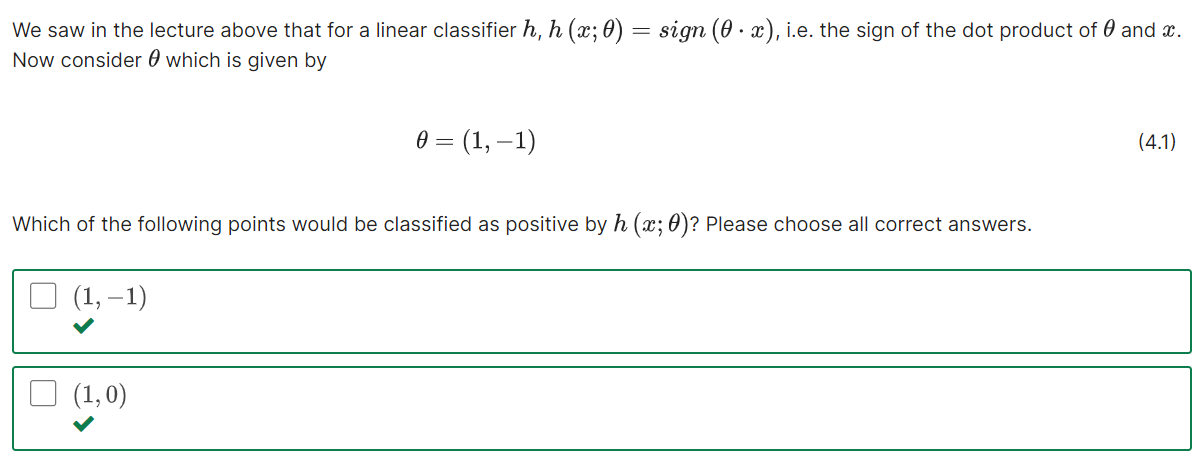


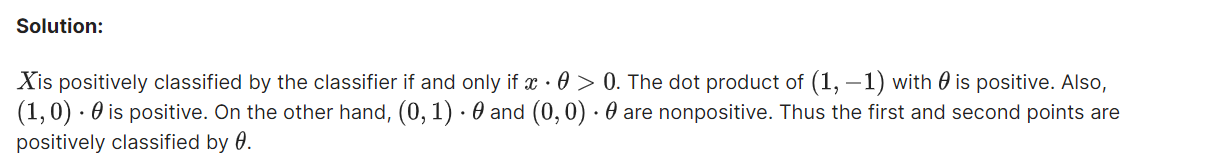
### 3. Linear Classifiers Mathematically Revisited

#### Inner product and Orthogonal vectors

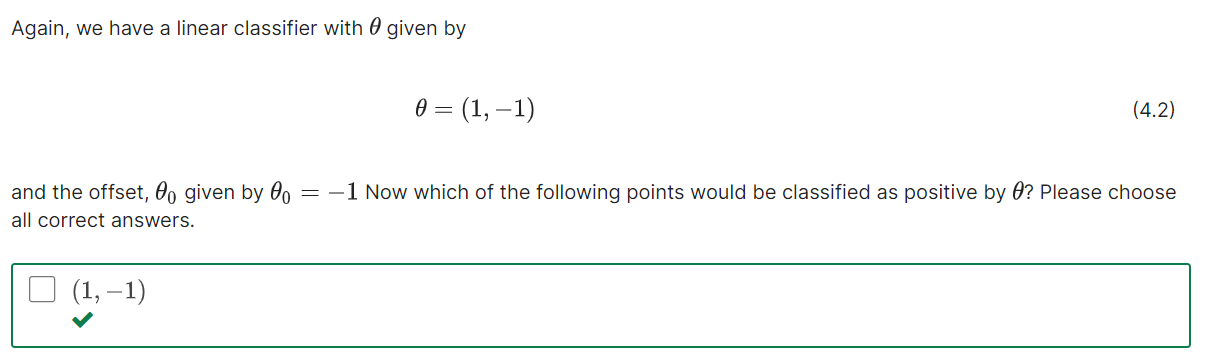


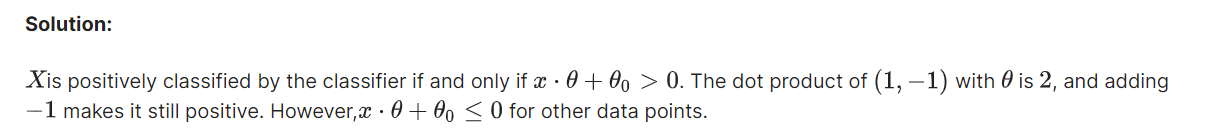
#### Linear Classifier Practice



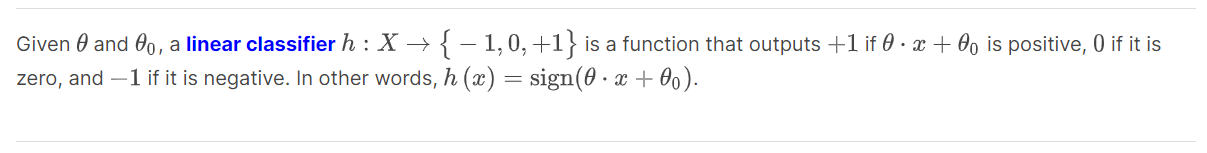


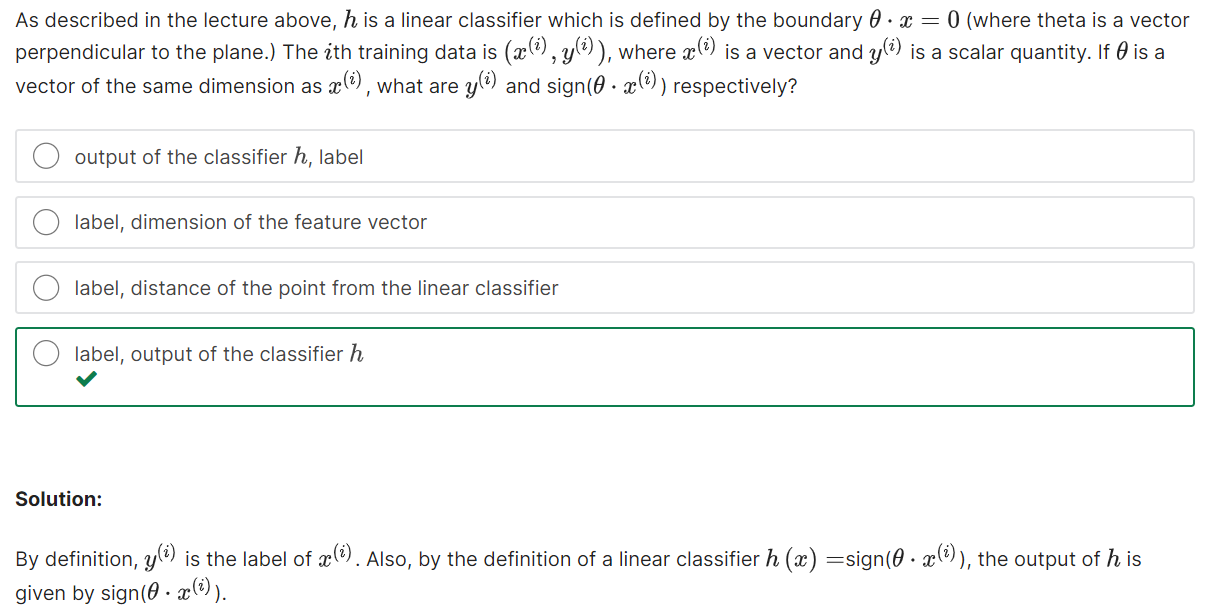
#### Offset Added





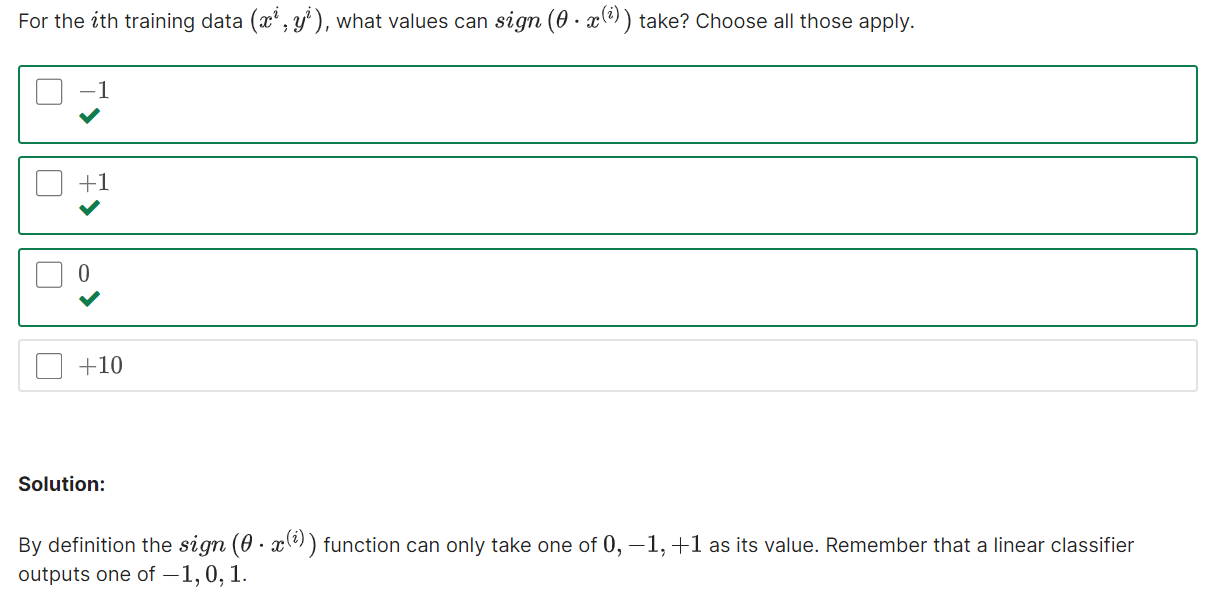
### 4. Linear Separation

**Question 1:**

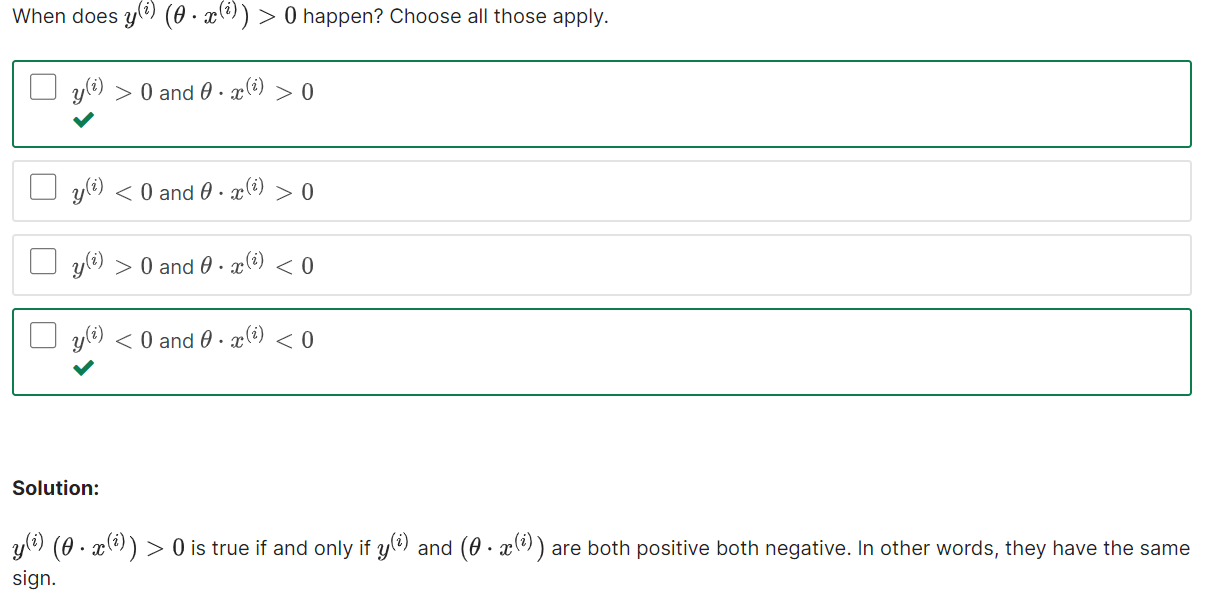


**Question 2:**

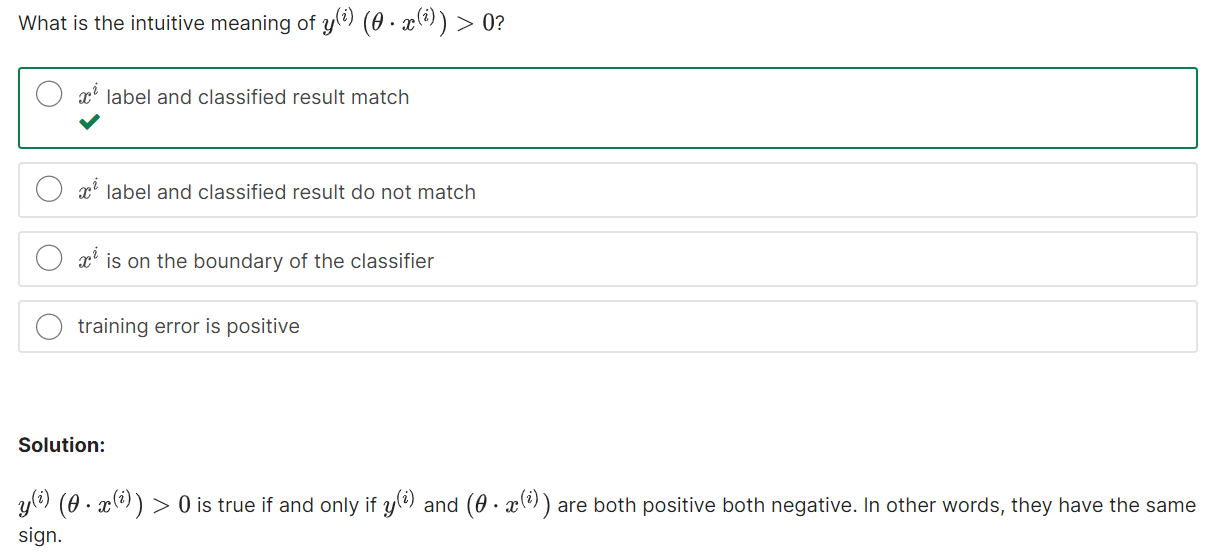
**Question 3:**



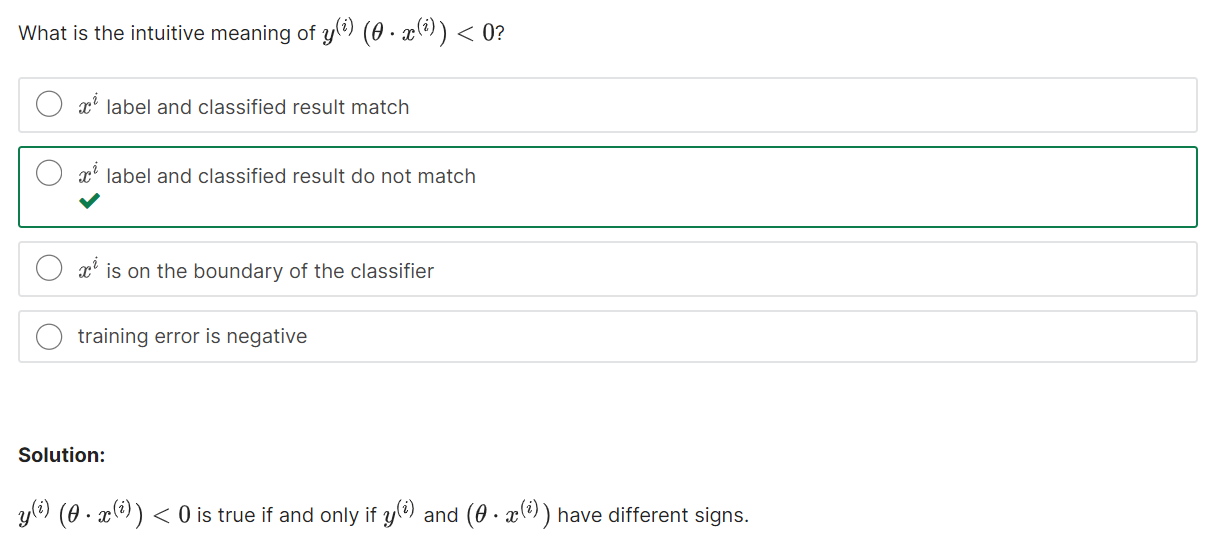
#### When the Product is Positive



#### Intuitive Meanings of Positive Product



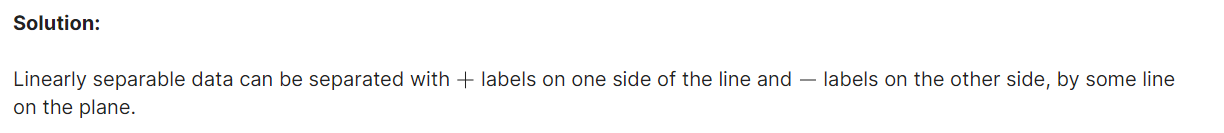
#### Intuitive Meanings of Negative Product



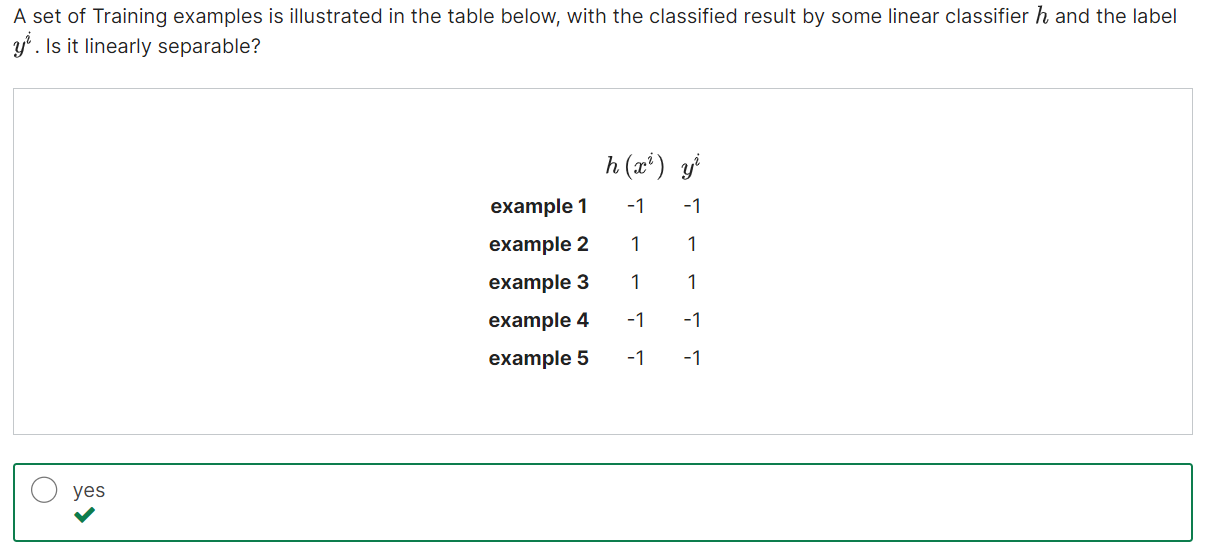
#### Linear Separation

**Question 1:**





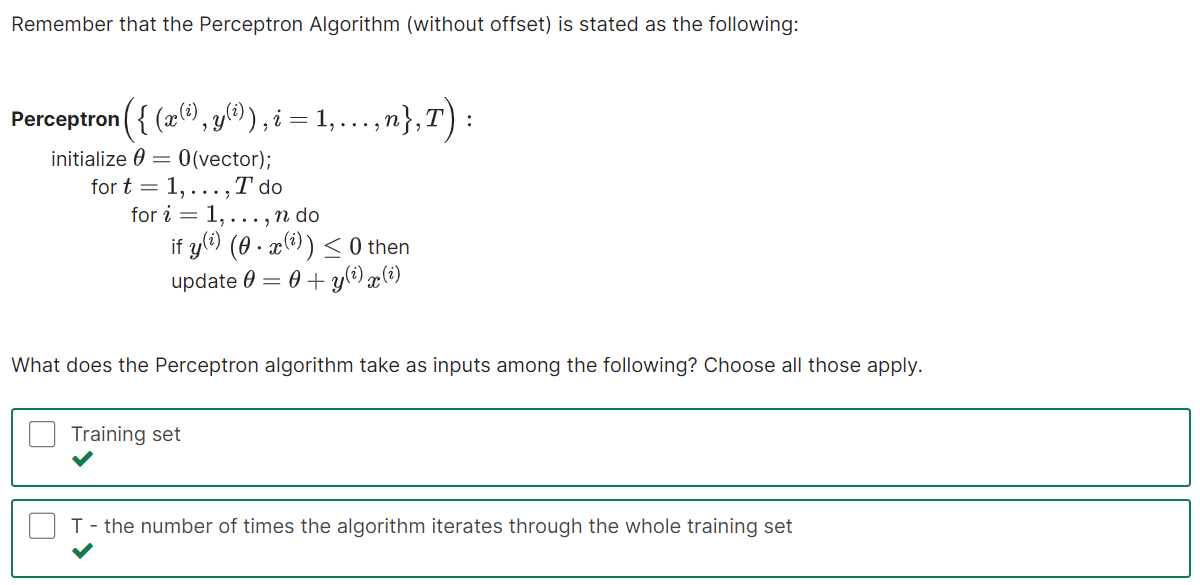
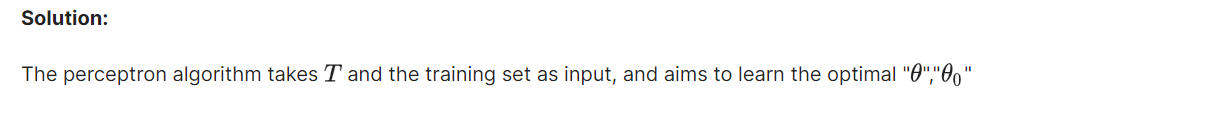
**Question 2:**



For linearly separable data, a linear classifier can perfectly separate the data. The provided classifier h(x) classifies all the given points correctly.

### 5. The Perceptron Algorithm

#### Perceptron Concept

#### Perceptron Update