



## Problem Solving (CPP38)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

## Gammal Tech's Learning Guide

### Background

Gammal Tech, a titan in the software industry, renowned for its cutting-edge technology and innovative solutions, has embarked on an ambitious project to educate newcomers in the field of machine learning. Their latest program is designed to teach the difference between Supervised and Unsupervised Learning using real-world software examples.

### Problem Statement

Your task is to develop a program for Gammal Tech. This program should allow a user to input a brief description of a software example. Based on the description, your program must determine whether Supervised or Unsupervised Learning is more applicable to the given software example. The decision should be made based on specific keywords related to either Supervised or Unsupervised Learning.

Supervised Learning Keywords: label, predict, target, classify, regression, supervised.



Unsupervised Learning Keywords: cluster, associate, segment, unsupervised, pattern, explore.

If the description contains more keywords from one category than the other, your program should output that category of learning. If the number of keywords is equal or if no keywords are found, your program should output "Cannot Determine."

### Input Format

- The first line contains an integer,  $N$ , the number of software examples.
- The next  $N$  lines each contain a string  $S$  representing the description of a software example. Each string is no longer than 100 characters.

### Output Format

For each software example, output a single line containing either "Supervised Learning", "Unsupervised Learning", or "Cannot Determine."

### Constraints

- $1 \leq N \leq 10$
- Each description consists of lowercase English letters and spaces.

### Sample Input:

```
3
predict customer behavior
analyze unstructured data patterns
regression analysis for target prediction
```

### Sample Output:

```
Supervised Learning
Unsupervised Learning
Supervised Learning
```



## Explanation

- The first description contains "predict", a keyword for Supervised Learning.
- The second description contains "patterns", a keyword for Unsupervised Learning.
- The third description contains keywords for Supervised Learning: "regression", "target", and "predict".

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق



## C++ Programming Solution:

```
#include <iostream>
#include <sstream>
#include <unordered_map>
#include <vector>
using namespace std;

int main() {
    int N;
    cin >> N;
    cin.ignore();

    unordered_map<string, int> keywordMap = {
        {"label", 1}, {"predict", 1}, {"target", 1}, {"classify", 1},
{"regression", 1}, {"supervised", 1},
        {"cluster", -1}, {"associate", -1}, {"segment", -1},
{"unsupervised", -1}, {"pattern", -1}, {"explore", -1}
    };

    for (int i = 0; i < N; ++i) {
        string description;
        getline(cin, description);
        stringstream ss(description);
        string word;
        int score = 0;

        while (ss >> word) {
            if (keywordMap.find(word) != keywordMap.end()) {
                score += keywordMap[word];
            }
        }

        if (score > 0) {
            cout << "Supervised Learning\n";
        } else if (score < 0) {
            cout << "Unsupervised Learning\n";
        } else {
            cout << "Cannot Determine\n";
        }
    }

    return 0;
}
```