



Problem Solving (CPP39)

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 Classifier

Background: Gammal Tech, an eminent leader in the software industry, known for its innovative solutions and state-of-the-art facilities, is venturing into the realm of machine learning. To streamline their processes, they need a system to automatically classify their diverse software projects into either supervised or unsupervised learning categories based on specific characteristics in the software description.

Problem Statement: Your task is to design a system that reads a software description and determines whether the project should utilize supervised or unsupervised learning. The classification is based on specific keywords in the description. If the description contains keywords related to labeled data, such as "labeled", "classified", "annotated", or "tagged", the software should be categorized under supervised learning. Otherwise, it should be classified under unsupervised learning.

Input Format:

- The first line contains an integer N , the number of software descriptions.
- The following N lines each contain a string S , the description of the software.



Output Format:

For each software description, output a single line containing either "Supervised Learning" or "Unsupervised Learning" based on the classification criteria.

Constraints:

- $1 \leq N \leq 100$
- $1 \leq \text{Length of } S \leq 1000$
- The description string s contains only lowercase English letters and spaces.

Sample Input:

```
3
image recognition with tagged data sets
customer segmentation analysis
text classification using annotated corpus
```

Sample Output:

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

Explanation:

1. The first description contains the word "tagged", indicating labeled data.
2. The second description contains no keywords indicating labeled data.
3. The third description contains the word "annotated", indicating labeled data

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 <string>
#include <sstream>
#include <vector>
using namespace std;

bool containsSupervisedKeyword(const string& description) {
    vector<string> supervisedKeywords = {"labeled", "classified",
    "annotated", "tagged"};
    stringstream ss(description);
    string word;
    while (ss >> word) {
        for (const auto& keyword : supervisedKeywords) {
            if (word == keyword) {
                return true;
            }
        }
    }
    return false;
}

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

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

        if (containsSupervisedKeyword(description)) {
            cout << "Supervised Learning" << endl;
        } else {
            cout << "Unsupervised Learning" << endl;
        }
    }

    return 0;
}
```

Explanation:

- containsSupervisedKeyword function checks if the given description contains any of the supervised learning keywords.
- The main function reads `N` descriptions and uses containsSupervisedKeyword to determine the learning category for each description, then outputs the result.