



Problem Solving (CPP41)

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'.

AI Algorithm Classifier at Gammal Tech

Background: Gammal Tech, a prestigious software development company, is known for its cutting-edge AI solutions. They are now developing a program to automatically classify AI algorithms needed for new software projects based on specific mathematical parameters.

Problem Statement: You are tasked with creating a program that classifies which AI algorithm should be used for each software project at Gammal Tech. The classification is based on three key mathematical parameters: Complexity (C), Data Size (D), and Real-Time Requirement (R). The AI algorithms to choose from are: Neural Networks, Decision Trees, Genetic Algorithms, and Simple Linear Regression.

- Neural Networks are chosen if Complexity is high (> 80) and Data Size is large (> 1000).
- Decision Trees are selected when Complexity is moderate (between 40 and 80) and Real-Time Requirement is not critical ($R < 50$).
- Genetic Algorithms are used for high Complexity (> 80) and critical Real-Time Requirement ($R \geq 50$).



- Simple Linear Regression is applied for low Complexity (≤ 40) regardless of other parameters.

Input Format:

- The first line contains an integer T , the number of test cases.
- For each test case, there is one line containing three space-separated integers: C , D , and R .

Output Format:

- For each test case, output a single line with the name of the AI algorithm chosen.

Constraints:

- $1 \leq T \leq 100$
- $0 \leq C, D, R \leq 100$

Sample Input:

```
3
85 1200 30
35 500 60
50 700 20
```

Sample Output:

```
Neural Networks
Simple Linear Regression
Decision Trees
```

Explanation:

- In the first case, Complexity and Data Size are high, so Neural Networks are chosen.
- In the second case, Complexity is low, so Simple Linear Regression is used.



- In the third case, Complexity is moderate with non-critical real-time requirement, so Decision Trees are selected.

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>
using namespace std;

string classifyAlgorithm(int C, int D, int R) {
    if (C > 80) {
        if (D > 1000) return "Neural Networks";
        if (R >= 50) return "Genetic Algorithms";
    } else if (C > 40 && C <= 80) {
        if (R < 50) return "Decision Trees";
    } else {
        return "Simple Linear Regression";
    }
    return "Unknown Algorithm"; // Fallback case
}

int main() {
    int T, C, D, R;
    cin >> T;
    while (T--) {
        cin >> C >> D >> R;
        cout << classifyAlgorithm(C, D, R) << endl;
    }
    return 0;
}
```