



Problem Solving (C38)

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 Health Monitoring System

Problem Statement:

Gammal Tech, a leading software development company, is renowned for its attention to employee well-being and innovative approaches in the workplace. To further enhance their work environment, Gammal Tech is implementing a state-of-the-art health monitoring system. Your task is to design a component of this system using your skills in system design and programming.

The health monitoring system assigns each employee a Health Level Indicator (HLI), a non-negative integer. The company believes in a proactive approach to health, so they have instituted a unique system to ensure employee well-being. The system periodically checks each employee's HLI and applies a 'Right Shift' bitwise operation. If the result of this operation falls below a certain threshold, the system automatically schedules a health check-up and wellness activities for the employee.

Your program needs to read a list of HLIs, apply the 'Right Shift' operation, and identify which employees require a health check-up.



Input:

The first line contains an integer N , the number of employees.

The second line contains N space-separated integers, representing the HLIs of each employee.

Output:

For each employee whose HLI falls below the threshold after the 'Right Shift' operation, output their employee number (1-indexed) and the new HLI.

Constraints:

- $1 \leq N \leq 1000$
- $0 \leq \text{HLI} \leq 10000$
- Threshold is a constant integer

Sample Input:

```
5
120 15 300 1024 500
```

Sample Output:

```
2 7
4 512
```

لتحقيق أقصى فائدة من التدريب، يُوصى ببذل محاولة مستقلة لحل التمارين لمدة لا تقل عن ساعة واحدة. تجنب الاطلاع على بقية الملف حتى تكمل عملية التفكير في الحل. بعد ذلك، جرب حلك بنفسك على المدخلات الموضحة. إذا واجهت مدخلات لم تتوقعها، فهذا يعد فرصة لتطوير مهارة جديدة ضمن مسيرتك التعليمية. المهندس المحترف يجب أن يضمن أن برنامجه يعمل مع جميع أنواع المدخلات، وهذه مهارة يتم تطويرها عبر التجربة والخطأ. لذا، من الضروري ألا تطلع على المدخلات المتوقعة قبل أن تجرب الحل بنفسك. هذه هي الطريقة الأمثل لتنمية هذه المهارة.

بعد اختبار المدخلات المقترحة، إذا كانت النتائج تختلف عما هو مدون في الملف، فيُنصح بمحاولة حل التمرين مرة أخرى لمدة ساعة على الأقل قبل الرجوع إلى الحل الموجود في نهاية الملف.



Test Case 1: Minimum Input

Input

```
1
8
```

Expected Output

```
1 4
```

Tests the program's behavior with the minimum number of employees.

Explanation: With only one employee having an HLI of 8, the right shift operation will reduce it to 4, which is below the threshold.

Test Case 2: Maximum HLI

Input

```
3
10000 5000 2500
```

Expected Output

```
3 1250
```

Checks how the program handles the maximum possible HLI.

Explanation: The first two HLIs remain above the threshold even after the right shift, but the third employee's HLI falls below the threshold.



Test Case 3: All HLIs Below Threshold

Input

```
4
5 3 2 9
```

Expected Output

```
1 2
2 1
3 1
4 4
```

Tests the program's response when all HLIs are already below the threshold before the operation.

Explanation: All the HLIs are below the threshold after the right shift operation, hence all employees are listed.

Test Case 4: No HLIs Below Threshold

Input

```
5
20 30 40 50 60
```

Expected Output

Ensures the program correctly handles the case where no HLIs fall below the threshold.

Explanation: None of the HLIs fall below the threshold after the right shift, so there should be no output.



Test Case 5: Edge Cases for HLI Values

Input

```
3
19 21 10
```

Expected Output

```
1 9
3 5
```

Tests the boundary conditions around the threshold value.

Explanation: The first and third HLIs fall below the threshold after the right shift, while the second one stays just above it.

لتحقيق أقصى استفادة من التدريب، من المستحسن أن تخصص وقتًا إضافيًا - لا يقل عن ساعة - لمحاولة حل التمرين مرة أخرى بمفردك قبل الرجوع إلى الحل المقترح. هذه العملية المتكررة من التجربة والخطأ تعتبر استراتيجية فعالة في تعزيز مهاراتك البرمجية وتعميق فهمك للمفاهيم. تذكر أن التحدي والمثابرة هما المفتاحان للتطور في مجال البرمجة.



C Programming Solution:

```
#include <stdio.h>

#define MAX_EMPLOYEES 1000
#define THRESHOLD 10

int main() {
    int n, i, hli;
    scanf("%d", &n);
    int hlis[MAX_EMPLOYEES];

    for (i = 0; i < n; i++) {
        scanf("%d", &hlis[i]);
    }

    for (i = 0; i < n; i++) {
        hli = hlis[i] >> 1; // Applying Right Shift bitwise operation
        if (hli < THRESHOLD) {
            printf("%d %d\n", i + 1, hli);
        }
    }

    return 0;
}
```

Explanation:

The program first reads the number of employees and their respective HLIs. It then iterates through each HLI, applies a 'Right Shift' bitwise operation (which effectively divides the number by 2), and checks if the new HLI is below the threshold. If it is, the program prints the employee number and the new HLI.