Assignment: Exhaustive Search

1. จากปัญหาจัดเก้าอี้ หากกำหนดให้เก้าอี้วางเรียงกัน n ตัว สำหรับเด็ก n คน โดยเด็กแต่ละคนแทนด้วยหมายเลข 1 ถึงหมายเลข n หากต้องการจัดเก้าอี้ให้เด็กทุกคน โดยมีเงื่อนไขว่าเด็กคนที่ 1 และเด็กคนที่ 2 ต้อง<u>นั่งติดกันเสมอ</u> จงเขียนโปรแกรมเพื่อแสดงวิธีจัดเก้าอี้ทั้งหมดตามเงื่อนไข

ข้อมูลนำเข้า n จำนวนเต็มแทนจำนวนเด็กและจำนวนเก้าอี้ที่ต้องการจัด โดยที่ 2 <= n <= 10

ข้อมูลส่งออก ลำดับที่นั่งของเด็ก n คน โดยเด็กคนที่ 1 และ 2 ต้องนั่งติดกันตลอดเวลา

ตัวอย่างข้อมูลนำเข้า	**ตัวอย่างข้อมูลส่งออก**
3	1 2 3 2 1 3 3 2 1 3 1 2
4	1 2 3 4 1 2 4 3 2 1 3 4 2 1 4 3 3 2 1 4 3 1 2 4 3 4 1 2 3 4 2 1 4 2 1 3 4 3 2 1 4 3 1 2 4 1 2 3

Assignment: Exhaustive Search

2. จากสไลด์หน้าที่ 23 จงปรับปรุงโค้ด subset1 เพื่อแสดงรายการเซตย่อยใน A ที่มีผลรวมเท่ากับ k โดยใช้วิธี Exhaustive search

ข้อมูลนำเข้า

บรรทัดที่ 1 n k จำนวนเต็มแทนจำนวนรายการจำนวนเต็ม และผลรวมที่ต้องการ คั่นด้วยช่องว่าง โดย

1 <= n, k <= 100

บรรทัดที่ 2 A1 A2 ... An รายการจำนวนเต็ม คั่นด้วยช่องว่าง โดยที่ 0 <= Ai <= 100

ข้อมูลส่งออก

แต่ละบรรทัดแสดงรายการเซตย่อยที่มีผลรวมเท่ากับ k แต่ละรายการเรียงลำดับตามสมาชิกใน A จากซ้ายไปขวา คั่นด้วยช่องว่าง

ตัวอย่างข้อมูลนำเข้า	**ตัวอย่างข้อมูลส่งออก**
5 12 25 10 9 2 1	10 2 9 2 1

3. กำหนดให้มีสินค้าทั้งหมด n ชิ้น แต่ละชิ้นมีมูลค่า v1, v2, ..., vn และน้ำหนัก w1, w2, ..., wn หากถุงรับน้ำหนักได้สูงสุด k จงเขียนโปรแกรม exhaustive search เพื่อแสดงวิธีการเลือกสินค้าใส่ถุงใบนี้เพื่อให้ได้รับมูลค่ารวม V สูงที่สุด

ข้อมูลนำเข้า

บรรทัดที่ 1 จำนวนเต็ม n k แทนจำนวนสินค้าและน้ำหนักรวมสูงสุดของถุง 1<= n<= 10,

1<= k <= 100

บรรทัดที่ 2 รายการ n จำนวนเต็มแทนมูลค่าสินค้า คั่นด้วยช่องว่าง โดยที่ 1<= vi <= 100

บรรทัดที่ 3 รายการ n จำนวนเต็มแทนน้ำหนักสินค้า คั่นด้วยช่องว่าง โดยที่ 1<= wi <= 100

ข้อมูลส่งออก

มูลค่ารวมสูงสุดที่สามารถจัดสินค้าใส่ถุงได้

ตัวอย่างข้อมูลนำเข้า	**ตัวอย่างข้อมูลส่งออก**
4 18 12 5 4 2 8 7 4 2	19

4. จงปรับปรุงโค้ด permutation ด้านล่างเพื่อแสดงวิธีวางควีนทั้งหมด n ตัวลงบนบอร์ดขนาด n*n โดยไม่เจอกันในทุกแนว

```c
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void permute(int X[], int start, int end) {
    if (start == end) {
        for (int i = 1; i <= end; i++) {
            printf("%d ", X[i]);
        }
        printf("\n");
        return;
    }

    for (int i = start; i <= end; i++) {
        swap(&X[start], &X[i]);
        permute(X, start + 1, end);
        swap(&X[start], &X[i]);
    }
}

int main() {
    int n = 3;
    int X[] = {-1, 1, 2, 3};

    permute(X, 1, n);

    return 0;
}
```

ข้อมูลนำเข้า

บรรทัดที่ 1 n จำนวนเต็มแทนจำนวนควีนที่ต้องการวาง โดยที่ 3 <= n <= 8

ข้อมูลส่งออก

แต่ละบรรทัดแสดงรายการแถว (row) ที่ใช้วางควีน 1 ถึง n ที่ไม่เจอกันในทุกทิศทาง คั่นด้วยช่องว่าง แต่ละรายการถูกเรียงลำดับตามแถวจากน้อยไปมาก แถวแรกเริ่มที่ 1

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
4	2 4 1 3 3 1 4 2

5. จงเขียนโปรแกรมแสดงวิธีการเรียงหนังสือ n เล่มบนชั้นหนังสือ โดยเรียงแบบ Lexicographic order

ข้อมูลนำเข้า

บรรทัดที่ 1 จำนวนเต็ม n แทนจำนวนหนังสือที่ต้องการเรียง โดยที่ 3 <= n <= 26

บรรทัดที่ 2 รายชื่อหนังสือ n เล่ม คั่นด้วยช่องว่าง (ชื่อหนังสือแสดงด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ตัวเดียว)

ข้อมูลส่งออก

แสดงวิธีการเรียงหนังสือที่เป็นไปได้

Input	Output
3 A B C	ABC ACB BAC BCA CAB CBA
4 E X A B	ABEX ABXE AEBX AEXB AXBE AXEB BAEX BAXE BEAX BEXA BXAE BXEA EABX EAXB EBAX EBXA EXAB EXBA XABE XAEB XBAE XBEA XEAB XEBA