DIETI UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Corso di Laurea Magistrale in Ingegneria Informatica

CORSO DI ALGORITMI E STRUTTURE DATI

Prof. ROBERTO PIETRANTUONO

Indicazioni

Si consegni un file in **formato .txt** nominandolo *CognomeNome*.txt, in cui è riportata l'implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell'algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore O(f(n))). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è O(n log n) al netto della complessità dell'algoritmo *x,* che è non nota"). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

PROBLEMA 1

Sia data una sequenza di numeri interi: $x_1, x_2, \ldots x_n$. Sia dato un numero intero S che rappresenta la somma due numeri della sequenza. Si scriva un algoritmo *divide et impera* per determinare gli elementi x*i* e x*j* la cui somma è uguale ad S.

Si assuma che:

- l'array non contenga duplicati;
- esistano sempre due numeri della sequenza la cui somma è S;
- possano esservi più soluzioni; nel qual caso va riportata la soluzione con la differenza $|x_i - x_j|$ minima.

INPUT

La prima riga dell'input riporta il numero T di casi di test. Seguono tre righe per ciascun caso di test: la prima riporta la dimensione N ($1 < N \leq 1000$) dell'array di input, la seconda riporta gli elementi del vettore, la terza riporta il valore della somma S.

OUTPUT

Per ogni caso di test, si stampino gli elementi x*i* e x*j* la cui somma è S.

Sample Input

```
2
40 40
80
5
10 2 6 8 4
10
```

Sample Output

```
40 40
4 6
```

PROBLEMA 2

La distanza di Hamming tra due stringhe di bit (interi binari) è il numero di posizioni di bit corrispondenti che differiscono. Questo può essere trovato usando XOR sui bit corrispondenti o, in modo equivalente, aggiungendo i bit corrispondenti (base 2) senza riporto. Ad esempio, nelle due stringhe di bit che seguono:

A	0 1 0 0 1 0 1 0 0 0
B	1 1 0 1 0 1 0 1 0 0
A XOR B =	1 0 0 1 1 1 1 1 0 0

La distanza di Hamming (H) tra queste stringhe a 10 bit è 6, il numero di 1 nella stringa XOR.
Si utilizzi il **backtracking.**

INPUT

L'input è costituito da diversi set di dati. La prima riga dell'input contiene il numero di set di dati ed è seguita da una riga vuota. Ogni set di dati contiene N, la lunghezza delle stringhe di bit e H, la distanza di Hamming, sulla stessa riga. C'è una riga vuota tra i test case.
Sia ssuma $1 \leq H \leq N \leq 16$.

OUTPUT

Per ogni set di dati stampa un elenco di tutte le possibili stringhe di bit di lunghezza N che si trovano alla distanza di Hamming H dalla stringa di bit contenente tutti gli 0 (origine). Cioè, tutte le stringhe di bit di lunghezza N con esattamente H 1 stampate in ordine lessicografico ascendente.

Sample Input

```
1
4 2
```

Sample Output

```
0011
0101
0110
1001
1010
1100
```