

# esercizi



*progetto CORDA  
informatica*

*alberto ferrari*

# permutazioni

- \* scrivere un programma in linguaggio C che stampi tutte le permutazioni ottenibili combinando le 4 lettere 'A', 'B', 'C', 'D'.
- \* segue un possibile output atteso del programma.

**ABCD BACD CABD DABC**

**ABDC BADC CADB DACB**

**ACBD BCAD CBAD DBAC**

**ACDB BCDA CBDA DBCA**

**ADBC BDAC CDAB DCAB**

**ADCB BDCA CDBA DCBA**

# criptoaritmetica

\* scrivere un programma in linguaggio C che trovi una soluzione al seguente problema di criptoaritmetica:

\*     C O C A +

\*     C O L A =

\*   -----

\* O A S I S

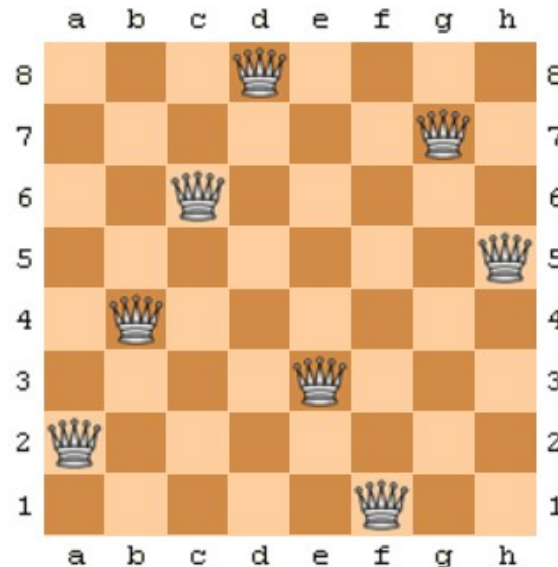
\* C,A,L,S,I ∈ [0..9] e O ∈ [1..9] , C≠O≠A≠L≠S≠I

# criptoaritmetica

- \* scrivere un programma in linguaggio C che trovi tutte le possibili soluzioni al seguente problema di criptoaritmetica:
- \* D U E +
- \* U N O =
- \* -----
- \* T R E
- \*  $D, U, E, N, O, T, R \in [0 \dots 9] \quad , \quad D \neq U \neq E \neq N \neq O \neq T \neq R$

# Problema delle 8 regine

- \* posizionare otto regine su una scacchiera 8x8 in modo che nessuna di esse possa catturarne un'altra
- \* nessuna regina deve avere una colonna, riga o diagonale in comune con un'altra regina.
- \* il problema è un esempio del più generale problema delle n regine su una scacchiera  $n \times n$
- \* [it.wikipedia.org/wiki/Rompicapo\\_delle\\_otto\\_regine](http://it.wikipedia.org/wiki/Rompicapo_delle_otto_regine)



# numeri semiprimi

- × un numero **semiprimo** è un intero  $\geq 2$  che si fattorizza come prodotto di due numeri primi (non necessariamente distinti)
- × esempi di semiprimi:
  - × 15, prodotto di 3 e 5
  - × 169, prodotto di 13 e 13
- × scrivere un programma che verifichi se un numero  $N$  è semiprimo
- × dati di input:
  - × il file **input.txt** contiene l'unico intero  $N$  , di cui si vuole verificare la semiprimalità
- × dati di output:
  - × il file **output.txt** contiene:
    - × i due primi che fattorizzano  $N$  , stampati su un'unica riga, in ordine non-decrescente, se  $N$  è semiprimo
    - × l'unico intero  $-1$  se  $N$  non è semiprimo.

# esempio di input/output in C

```
#include <stdio.h>

int main() {
    FILE *fin = fopen("input.txt", "r");
    FILE *fout = fopen("output.txt", "w");
    unsigned n;
    fscanf(fin, "%u", &n);
    ...
    fprintf(fout, "%d", ...);
    fclose(fin);
    fclose(fout);
    return 0;
}
```

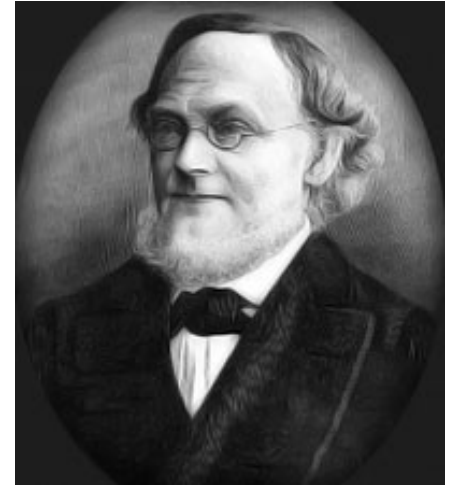
# numeri primi sexy

- \* in matematica due numeri primi si dicono sexy quando la loro differenza è uguale a sei, ovvero formano coppie di tipo:
  - \*  $(p, p + 6)$
- \* il nome di queste coppie di numeri primi deriva dalla parola latina sex (ovvero sei).
- \* scrivere un programma che visualizza tutte le coppie di numeri sexy  $(p_1, p_2)$  con  $p_1 < 100$  e  $p_2 < 100$



# Goldbach

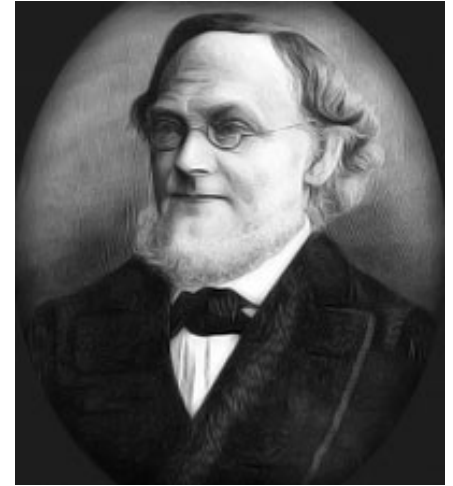
- × la congettura di Goldbach è uno dei più vecchi problemi irrisolti nella teoria dei numeri
- × essa afferma che ogni numero pari maggiore di 2 può essere scritto come somma di due numeri primi (che possono essere anche uguali)
- × scrivere un programma che riceve un numero intero pari **n** e visualizza la coppia di numeri primi **p1** e **p2** tali che **p1+p2 = n**



*Christian Goldbach (Königsberg, 18 marzo 1690 – Mosca, 20 novembre 1764)  
è stato un matematico tedesco, molto noto per la sua congettura sui numeri primi formulata nel 1742 e ancora aperta.*

# Goldbach debole

- × la congettura debole di Goldbach asserisce che tutti i numeri dispari maggiori di 7 possono essere scritti come somma di tre primi
- × scrivere un programma che riceve un numero intero dispari **n** e visualizza 3 numeri primi **p1**, **p2**, **p3** tali che  **$p1+p2+p3 = n$**  ...
- × oppure confuta la congettura debole di Goldbach :)



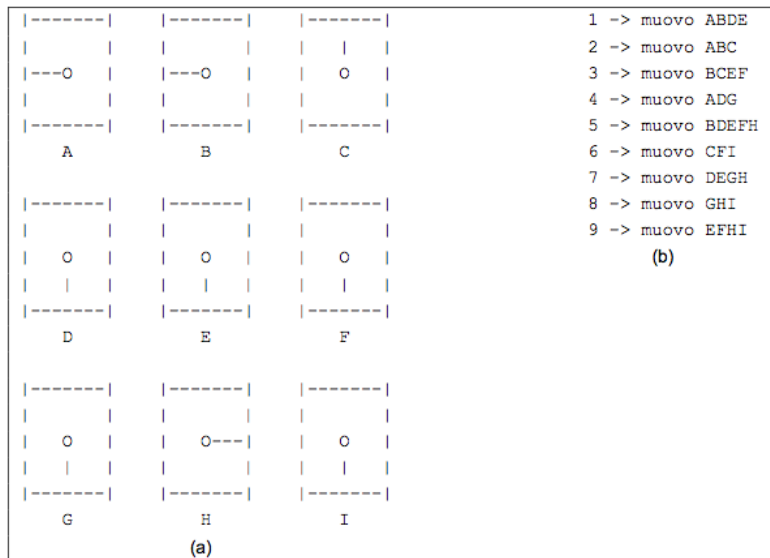
*Christian Goldbach (Königsberg, 18 marzo 1690 – Mosca, 20 novembre 1764)  
è stato un matematico tedesco, molto noto per la sua congettura sui numeri primi formulata nel 1742 e ancora aperta.*

# insiemi uguali

- \* si ricevono in input **n** numeri interi con valore compreso fra **1** e **m**, poi se ne ricevono altri **n** (sempre con valore tra **1** e **m**).
- \* verificare che l'insieme formato dai **primi n** (senza che i numeri vengano ripetuti più volte) sia uguale all'insieme dei **secondi n** (senza che i numeri vengano ripetuti più volte)

# orologi

- × si considerino nove orologi in una matrice 3x3 che vanno tutti portati a segnare le ore 12
- × per portare spostare le lancette degli orologi si hanno a disposizione nove mosse
  - × ogni mossa corrisponde allo spostamento della lancetta di qualche orologio in senso orario di 90 gradi
  - × ad esempio, la mossa 5 sposta in senso orario di 90 gradi le lancette degli orologi B, D, E, F e H.
- × l'obiettivo finale è fare in modo che tutti gli orologi segnino le 12 applicando il minor numero di mosse
- × l'input fornito dall'utente è sempre di nove cifre. Ogni cifra identifica la posizione iniziale delle lancette: 0=ore 12, 1=ore 3, 2=ore 6, 3=ore 9
- × l'output del programma è la più breve sequenza di mosse che riportano tutti gli orologi alle 12. Se ci sono più soluzioni se ne deve comunque presentare una sola



- 1 -> muovo ABDE
  - 2 -> muovo ABC
  - 3 -> muovo BCEF
  - 4 -> muovo ADG
  - 5 -> muovo BDEFH
  - 6 -> muovo CFI
  - 7 -> muovo DEGH
  - 8 -> muovo GHI
  - 9 -> muovo EFHI
- (b)

L'esempio corrisponde all'input: 3 3 0 2 2 2 2 1 2

L'output potrebbe essere 5 8 4 9