

Problemi pecuniari (pinball)

Durante una sessione di gioco al famosissimo locale *QN32* Andrea, in seguito ad una scommessa di troppo, si ritrova senza *Soldi*¹. Nonostante la terribile situazione, si convince che il miglior modo per recuperare il suo capitale sia ignorare i consigli dei suoi amici e continuare a giocare, ma ad un gioco diverso. Il gioco da lui scelto è *Pinball 2D*².



Figura 1: *Pinball 2D*, momenti dopo il disastro.

Il gioco funziona così: ci sono N paletti disposti su un piano cartesiano infinito alle coordinate (x_i, y_i) , per ogni $i = 0 \dots N - 1$. I paletti possono essere alzati o abbassati. L'obiettivo del gioco è scegliere un insieme di paletti da alzare e formarci un poligono non intrecciato. Al giocatore verranno forniti tanti *Soldi* quanti sono i **punti a coordinate intere** contenuti dal poligono, esclusi quelli che appartengono al perimetro della figura.

Nota che scelto un insieme di K paletti da alzare si possono formare $\frac{K!}{2K}$ poligoni. Sta al giocatore scegliere quello che massimizza i *Soldi* vinti!

Visto l'evidente "*Skill Issue*" di Andrea in geometria, aiutalo a trovare il massimo capitale di *Soldi* che può vincere! Dato che questo valore può essere enorme, dovrai restituire la risposta **modulo 1 000 000 007**.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

📖 Tra gli allegati a questo task troverai un template `pinball.cpp` con un esempio di implementazione.

¹Una peculiare valuta internazionale, 1 *Soldo* \approx 1.18€.

²Antenato del gioco Toscano "*Calcetto 3D*".

Il file di input è composto da $N + 1$ righe:

- Riga 1: l'intero N .
- Riga 2 \dots $N + 1$: le coordinate x_i e y_i .

Il file di output è composto da 1 riga:






- Riga 1: la risposta al problema.

Assunzioni

- $1 \leq N \leq 1\,000\,000$.
- $-1\,000\,000 \leq x_i, y_i \leq 1\,000\,000$.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

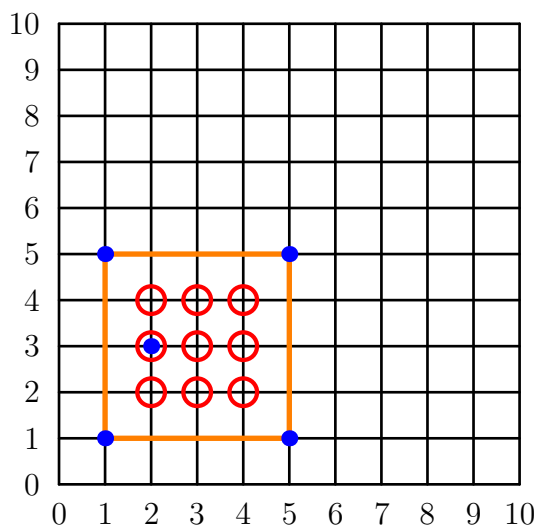
- **Subtask 1** (0 punti) Casi d'esempio.

- **Subtask 2** (30 punti) Gli insiemi di tutte le x_i e le y_i hanno entrambi cardinalità 2.

- **Subtask 3** (50 punti) $N \leq 10$.

- **Subtask 4** (5 punti) $-10\,000 \leq x_i, y_i \leq 10\,000$.

- **Subtask 5** (15 punti) Nessuna limitazione aggiuntiva.


Esempi di input/output

stdin	stdout
5 1 1 1 5 2 3 5 1 5 5	9
8 8 2 6 1 1 4 8 3 9 5 1 3 8 3 7 2	18

Spiegazione

Nel primo esempio sono evidenziati in blu i paletti, in arancione il perimetro del poligono e in rosso i punti compresi nella figura. Con questa configurazione, il giocatore vince 9 *Soldi* e si può dimostrare che non esiste soluzione migliore. Il paletto a coordinate (2,3) è abbassato, e viene contato come punto contenuto dal poligono.



Nel secondo caso di esempio, si può dimostrare come il poligono mostrato in figura sia quello che massimizza i *Soldi* vinti.

