

Emoji di compleanno (compleanno)

Come spesso accade, tra un buongiorno ed un caffè, anche oggi William ha ricevuto una notifica da Facebook per ricordargli del compleanno di uno dei suoi amici! Per festeggiarlo, vuole quindi postare sulla sua bacheca un messaggio d'auguri, una simpatica GIF, e tante piccole torte di compleanno... proprio come questa: 🎂

Inoltre, per far meglio capire quanto i suoi auguri siano sentiti, William inserisce sempre un numero di emoji pari al numero di anni compiuti.



Per esempio, se la persona in questione oggi compie 18 anni, il testo del suo messaggio sarà:

“🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂”

Per risparmiare tempo, però, William vuole scrivere queste 18 emoji nel modo più efficiente possibile. Sul suo computer, infatti, ci sono diversi modi per costruire questo messaggio: un primo modo (piuttosto lento) è quello di scrivere per 18 volte l'emoji; un modo più efficiente, invece, si ottiene sfruttando in modo furbo la funzione “copia incolla”. Più precisamente, ci sono 3 operazioni disponibili:

- Inserire un'emoji 🎂 dalla lista di emoji fornite da Facebook. L'operazione richiede **1 click**.
- Usare il mouse per selezionare il testo, fare un click col tasto destro del mouse e selezionare “Copia”, salvando nella clipboard tutte le emoji scritte finora. Questa operazione richiede **2 click**.
- Fare click col destro e selezionare “Incolla”, aggiungendo in fondo al messaggio le emoji salvate nella clipboard. L'operazione richiede **1 click**.

Diciamo che un modo di scrivere le emoji è **ottimale** quando non è possibile scriverle facendo meno click. Dato che Facebook sta cominciando a mandare un po' troppe notifiche al giorno per i vari compleanni (colpa di tutte le nuove richieste di amicizia che arrivano dai gruppi dedicati ai gattini), William ha deciso di scrivere un programma che implementi la strategia ottimale al posto suo!

Possono esistere più strategie ottimali equivalenti, vale a dire: ci possono essere tanti modi diversi per scrivere le emoji con il minor numero possibile di operazioni. È sufficiente che il tuo programma implementi una qualsiasi strategia ottimale.

Implementazione

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

📎 Tra gli allegati a questo task troverai un template `compleanno.c` e `compleanno.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

```
C/C++ void auguri(long long int N);
```

- L'intero N rappresenta il numero di anni compiuti dall'amico su Facebook.

Il tuo programma potrà utilizzare le seguenti funzioni, definite nel grader:

C/C++	<pre>void aggiungi(); void copia(); void incolla();</pre>
-------	---

- Chiamare `aggiungi()` equivale a fare 1 click sull'emoji 🎂.
- Chiamare `copia()` equivale a fare 2 click: uno per selezionare tutto, ed uno per copiare.
- Chiamare `incolla()` equivale a fare 1 click per incollare.

Il grader chiamerà la tua funzione `auguri(N)`, passando come parametro il numero di anni compiuti. La tua funzione potrà chiamare le tre funzioni `aggiungi()`, `copia()`, e `incolla()`. Al termine, il numero totale di click effettuati verrà registrato dal grader sul file di output.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che puoi usare per testare le tue soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama le funzioni che devi implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da una riga contenente l'unico intero N , il numero di anni compiuti.

Il file di output è composto da una riga contenente un unico intero, il numero totale di click effettuati.

Assunzioni

- $1 \leq N \leq 10^{18}$.

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]:** Esempi.
- **Subtask 2 [7 punti]:** N è una potenza di 2, ovvero: $N = 2^k$ per un certo k .
- **Subtask 3 [6 punti]:** $N \leq 50$.
- **Subtask 4 [5 punti]:** $N \leq 800$.
- **Subtask 5 [19 punti]:** $N \leq 10\,000$.
- **Subtask 6 [17 punti]:** $N \leq 500\,000$.
- **Subtask 7 [22 punti]:** $N \leq 10^9$.
- **Subtask 8 [8 punti]:** $N \leq 10^{12}$.
- **Subtask 9 [15 punti]:** $N \leq 10^{15}$.
- **Subtask 10 [1 punto]:** Nessuna limitazione specifica.

Esempi di input/output

stdin	stdout
8	7
18	10

Spiegazione

Il **primo caso di esempio** si può risolvere così:

1. Fai click sull'emoji. Il testo diventa: 🎂
2. Fai di nuovo click sull'emoji. Il testo diventa: 🎂🎂
3. Fai di nuovo click sull'emoji. Il testo diventa: 🎂🎂🎂
4. Fai di nuovo click sull'emoji. Il testo diventa: 🎂🎂🎂🎂
5. Usa il mouse per selezionare tutto ciò che hai scritto.
6. Fai click col tasto destro del mouse e seleziona “Copia”.
7. Fai click col tasto destro del mouse e seleziona “Incolla”. **Finito!**

Il **secondo caso di esempio** si può risolvere così:

1. Fai click sull'emoji. Il testo diventa: 🎂
2. Fai di nuovo click sull'emoji. Il testo diventa: 🎂🎂
3. Fai di nuovo click sull'emoji. Il testo diventa: 🎂🎂🎂
4. Usa il mouse per selezionare tutto ciò che hai scritto.
5. Fai click col tasto destro del mouse e seleziona “Copia”.
6. Fai click col tasto destro del mouse e seleziona “Incolla”. Il testo diventa: 🎂🎂🎂🎂🎂🎂
7. Fai di nuovo click col destro e seleziona “Incolla”. Il testo diventa: 🎂🎂🎂🎂🎂🎂🎂🎂🎂🎂
8. Usa il mouse per selezionare tutto ciò che hai scritto.
9. Fai click col tasto destro del mouse e seleziona “Copia”.
10. Fai click col tasto destro del mouse e seleziona “Incolla”. **Finito!**