



## Circuiti bruciati (circuiti)

Il server delle Olimpiadi Italiane di Informatica si è guastato proprio il giorno della finale nazionale! Ce la faranno i sistemisti molisani a riportarlo online prima dell'inizio della gara?

Dopo un'attenta analisi, è venuto fuori che ci sono dei circuiti bruciati da sostituire. Fortunatamente, nel laboratorio del Marconi ci sono un saldatore ed abbastanza componenti per ricostruire da zero i circuiti mancanti. Dario, il tutor smanettone, si è offerto di eseguire la riparazione. Quello che gli manca però è il progetto (schema elettrico) dei circuiti. Aiuta Dario a progettare lo schema dei circuiti mancanti!

Un circuito legge un array di  $N$  interi tramite i *registri di input*  $\text{in}[i]$  ( $0 \leq i < N$ ), e può scrivere il risultato sui *registri di output*  $\text{out}[i]$ . Ci sono tre tipi di circuiti da sostituire, che devono realizzare le seguenti operazioni.

<b>Somma</b>	Scrivere su $\text{out}[0]$ la somma degli $N$ interi in input.
<b>Prefissi</b>	Scrivere sui registri di output l'array delle somme prefisse, definito come $\text{out}[i] = \text{in}[0] + \dots + \text{in}[i]$ con $0 \leq i < N$ .
<b>Massimo sottoarray</b>	Scrivere su $\text{out}[0]$ il più grande numero ottenibile come somma di elementi consecutivi dell'array di input (ovvero, che si possa scrivere come $\text{in}[i] + \dots + \text{in}[j]$ con $0 \leq i \leq j < N$ ). Anche il vettore vuoto vale come sottoarray, per cui il risultato di questa operazione è sempre almeno 0.

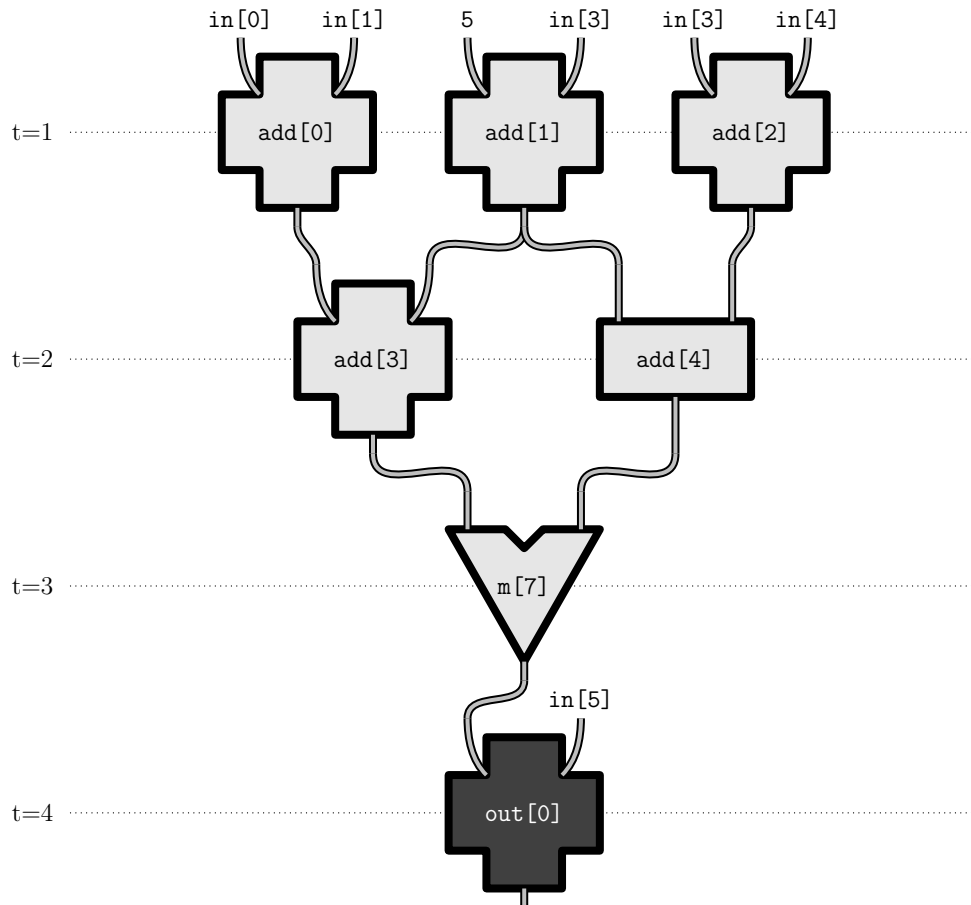
Oltre ai registri di input e output, un circuito può contenere *registri intermedi*. Inoltre esistono dei registri speciali che forniscono delle *costanti* che possono essere usate come ingresso per gli altri componenti.

Un circuito è formato da diversi *componenti*, come esemplificato in Figura 1. Ogni componente è collegata a due registri di ingresso e un registro di uscita, e implementa una fra le seguenti operazioni matematiche: somma (+), differenza (-) e massimo (max). Appena vengono scritti i valori di entrambi i registri di ingresso, il componente effettua il calcolo e, dopo un ciclo di clock, scrive il risultato dell'operazione scelta sul suo registro di uscita. Il registro di uscita può a sua volta essere collegato (come registro di ingresso) ad uno o più componenti, che eseguiranno il calcolo appena avranno disponibili entrambi i loro ingressi e così via.

I registri di input possono essere usati solo come ingressi, mentre i registri intermedi e i registri di output sono uscite di **esattamente una** componente. Il calcolo termina quando tutti i registri sono stati scritti.

Per ciascun circuito che deve essere progettato, conosci il tipo di circuito (somma, prefissi o massimo sottoarray), il numero  $N$  di registri di input, e il tempo di calcolo  $C$  desiderato (espresso in cicli di clock).

Aiuta Dario a progettare tutti i circuiti che mancano, in modo tale che funzionino **qualunque siano i dati in ingresso**, e che il loro tempo di calcolo sia il più possibile vicino a quello desiderato. Cerca inoltre di non usare troppi componenti: ogni circuito viene con 1100 componenti in regalo, ma ogni ulteriore componente va comprato!



Esempio di realizzazione della formula:

$$\text{out}[0] = \max(\text{in}[0] + \text{in}[1] + 5 + \text{in}[3], (5 + \text{in}[3]) - (\text{in}[3] + \text{in}[4])) + \text{in}[5]$$

I circuiti che dovrai progettare sono i seguenti:

Input	Tipo	$N$	$C$
input000.txt	Somma	7	3
input001.txt	Somma	256	8
input002.txt	Somma	1093	11
input003.txt	Prefissi	7	5
input004.txt	Prefissi	128	13
input005.txt	Prefissi	371	15
input006.txt	Massimo sottoarray	7	15
input007.txt	Massimo sottoarray	64	31
input008.txt	Massimo sottoarray	110	35
input009.txt	Massimo sottoarray	124	35

## Formato dell'input

I circuiti da progettare sono riportati nei file di input come nella tabella precedente. Ogni file consiste in tre interi  $T$ ,  $N$  e  $C$  dove  $T$  indica il tipo di circuito da realizzare (1 per la somma, 2 per i prefissi, 3 per il massimo sottoarray).

## Formato dell'output

Il circuito da te progettato va descritto sotto forma di componenti e corrispettive connessioni. Ogni riga del file di output rappresenta un componente, secondo uno dei tre seguenti formati possibili:

$$C[k] = A[i] \text{ op } B[j] \quad C[k] = i \text{ op } B[j] \quad C[k] = A[i] \text{ op } j$$

in cui:

- $A$ ,  $B$  e  $C$  sono nomi di array. Questi nomi possono essere composti da massimo 50 caratteri alfanumerici (maiuscoli o minuscoli), e non devono cominciare con una cifra.
- $i$ ,  $j$ ,  $k$  sono costanti intere positive, che devono stare in un intero a 32 bit con segno.
- $\text{op}$  è un operatore a scelta fra  $+$ ,  $-$  o  $\text{max}$ .

Tale riga definisce un componente che ha  $A[i]$  e  $B[j]$  come registri di ingresso (o le costanti  $i$  e  $j$  rispettivamente),  $C[k]$  come registro di uscita, e implementa l'operazione  $\text{op}$ .

I due array speciali `in` e `out` indicano rispettivamente i registri di input e output. È possibile usare a piacere altri array come registri intermedi. Tuttavia, prima di definire un componente che usa un certo registro intermedio come ingresso, è necessario che un componente **precedentemente definito** nel file lo abbia come uscita.

## Assegnazione del punteggio

I file di output inviati verranno controllati e valutati da un correttore il quale verificherà la correttezza del formato e la correttezza del circuito realizzato, e assegnerà a ciascuno di essi un punteggio da 0 a 10.

Se il file inviato non è corretto riceverà punteggio 0. Altrimenti, gli verrà assegnato un punteggio  $P$  secondo la seguente formula:

$$P = 10 \cdot \min \left\{ 1, \frac{C_{\text{richiesto}}}{C_{\text{ottenuto}}} \right\} \cdot \min \left\{ 1, \frac{1100}{S_{\text{ottenuto}}} \right\}$$

dove  $C_{\text{ottenuto}}$  è il tempo di calcolo del tuo circuito, e  $S_{\text{ottenuto}}$  è il suo numero di componenti. Nota che il tuo file riceverà punteggio pieno 10 se e solo se  $C_{\text{ottenuto}}$  è pari o inferiore a  $C_{\text{richiesto}}$  e  $S_{\text{ottenuto}}$  è pari o inferiore a 1100.

## Esempi di input/output

Lo schema disegnato nel testo corrisponde al seguente file di output:

```
add[0] = in[0] + in[1]
add[1] = 5 + in[3]
add[2] = in[3] + in[4]
add[3] = add[0] + add[1]
add[4] = add[1] - add[2]
m[7] = add[3] max add[4]
out[0] = m[7] + in[5]
```

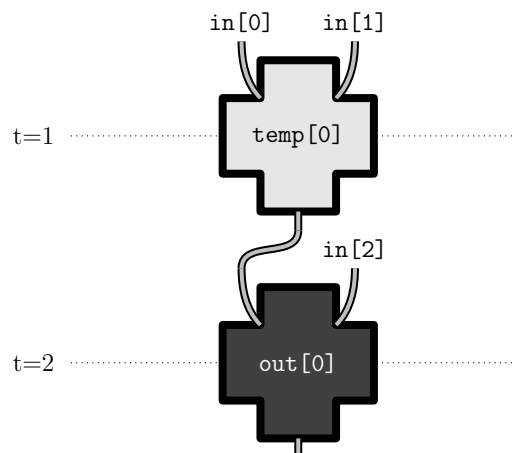
anche se non è la soluzione di nessun input valido. La visualizzazione di questo output, **o di qualunque altro da te prodotto**, può essere ottenuta cliccando sull'icona di un circuito presente nella barra delle applicazioni alla sinistra del tuo schermo. L'applicazione corrispondente ti chiederà di selezionare un file e ti mostrerà il PDF ad esso corrispondente (contenente il disegno di un circuito o eventuali errori presenti nel file). Il visualizzatore posiziona i componenti corrispondenti allo stesso ciclo di clock  $t$  nell'ordine in cui sono riportati nel file, da sinistra verso destra.

Possibili soluzioni di input, diversi da quelli che dovrai progettare, sono presentati di seguito.

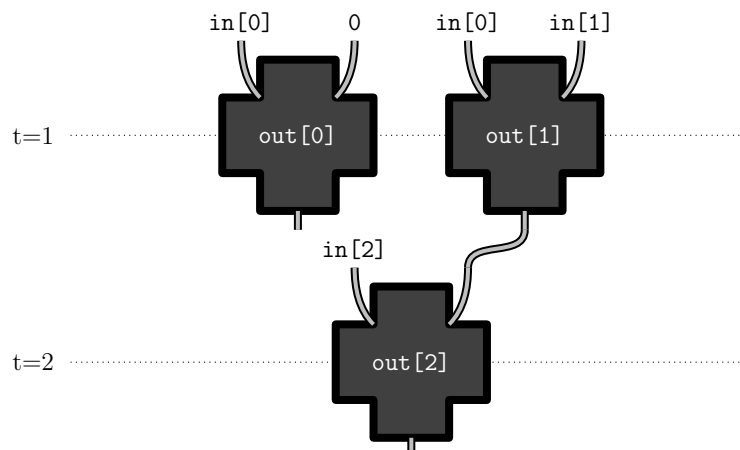
input	output
1 3 2	$temp[0] = in[0] + in[1]$ $out[0] = temp[0] + in[2]$
2 3 4	$out[0] = in[0] + 0$ $out[1] = in[0] + in[1]$ $out[2] = in[2] + out[1]$
3 3 7	$somma[0] = 0 + 0$ $somma[12] = in[0] + in[1]$ $somma[23] = in[1] + in[2]$ $mx[0] = somma[0] \max in[0]$ $somma[123] = somma[12] + in[2]$ $mx[1] = mx[0] \max in[1]$ $mx[2] = mx[1] \max in[2]$ $mx[3] = somma[12] \max mx[2]$ $mx[4] = mx[3] \max somma[23]$ $out[0] = mx[4] \max somma[123]$

## Spiegazione

Nel **primo caso di esempio** si tratta di sommare 3 interi in input, per esempio, se l'input fosse  $in[0] = -10$ ,  $in[1] = 3$ ,  $in[2] = 5$  allora il valore di  $temp[0]$  sarà  $-7$  e  $out[0]$  varrà  $-2$ . Le due operazioni devono essere eseguite una dopo l'altra, impiegando quindi due cicli di clock a produrre il risultato.



Nel **secondo caso di esempio**, dato lo stesso input del caso precedente, l'output varrà:  $out[0] = -10$ ,  $out[1] = -7$ ,  $out[2] = -2$ . Nota che il terzo componente deve attendere il risultato del secondo, quindi anche in questo caso il circuito impiega 2 cicli di clock.



Nel **terzo caso di esempio**, dato sempre lo stesso input, i valori dei registri saranno:

somma[0] = 0  
 somma[12] = -7  
 somma[23] = 8  
 somma[123] = -2

mx[0] = 0  
 mx[1] = 3  
 mx[2] = 5  
 mx[3] = 5

mx[4] = 8  
 out[0] = 8

