



Guida per l'utente

VisualBooth

UN APPLICATIVO GRAFICO PER COMPRENDERE IL FUNZIONAMENTO
DELL'ALGORITMO DI BOOTH E NON SOLO.



Release & Installation Notes

L'applicativo si pone l'obiettivo di presentare visivamente il funzionamento del noto Algoritmo di Booth facendo uso di un'interfaccia grafica completa sviluppata in Java, in ambiente Processing, resa disponibile al download sia per OS Windows che Linux. Si rimanda alle note di installazione e note di rilascio per le relative istruzioni di installazione e avvio. Le operazioni sono da intendersi a lunghezza fissa di 8 bit.

Prodotto Naturale

- ▶ Trattasi del più semplice dei prodotti, analogamente all'algoritmo utilizzato per i numeri decimali:
- ▶ Procedendo dalla cifra meno significativa alla più significativa del secondo fattore si effettua il prodotto della cifra presa in esame con le cifre del primo fattore, andando così a generare il primo risultato nel passo 1.
- ▶ Il risultato ottenuto dunque sarà la ripetizione del primo fattore se la cifra presa in questione del secondo fattore è 1, oppure una serie di zeri se la cifra è 0.
- ▶ L'operazione si ripete per ogni cifra del secondo fattore.
- ▶ Il risultato è calcolato sommando per colonna le cifre di tutti i passi, tenendo in considerazione il riporto

Algoritmo di Booth

- ▶ L'**algoritmo del prodotto di Booth**, o semplicemente **algoritmo di Booth**, è un algoritmo per il calcolo del prodotto tra due numeri binari con segno, espressi nella notazione complemento a due. Fu inventato dal fisico Andrew Donald Booth nel 1951, originariamente allo scopo di velocizzare i calcoli necessari a una ricerca che Booth stava svolgendo nel settore della cristallografia, avendo a disposizione una calcolatrice lenta nelle somme ma veloce nello shift.

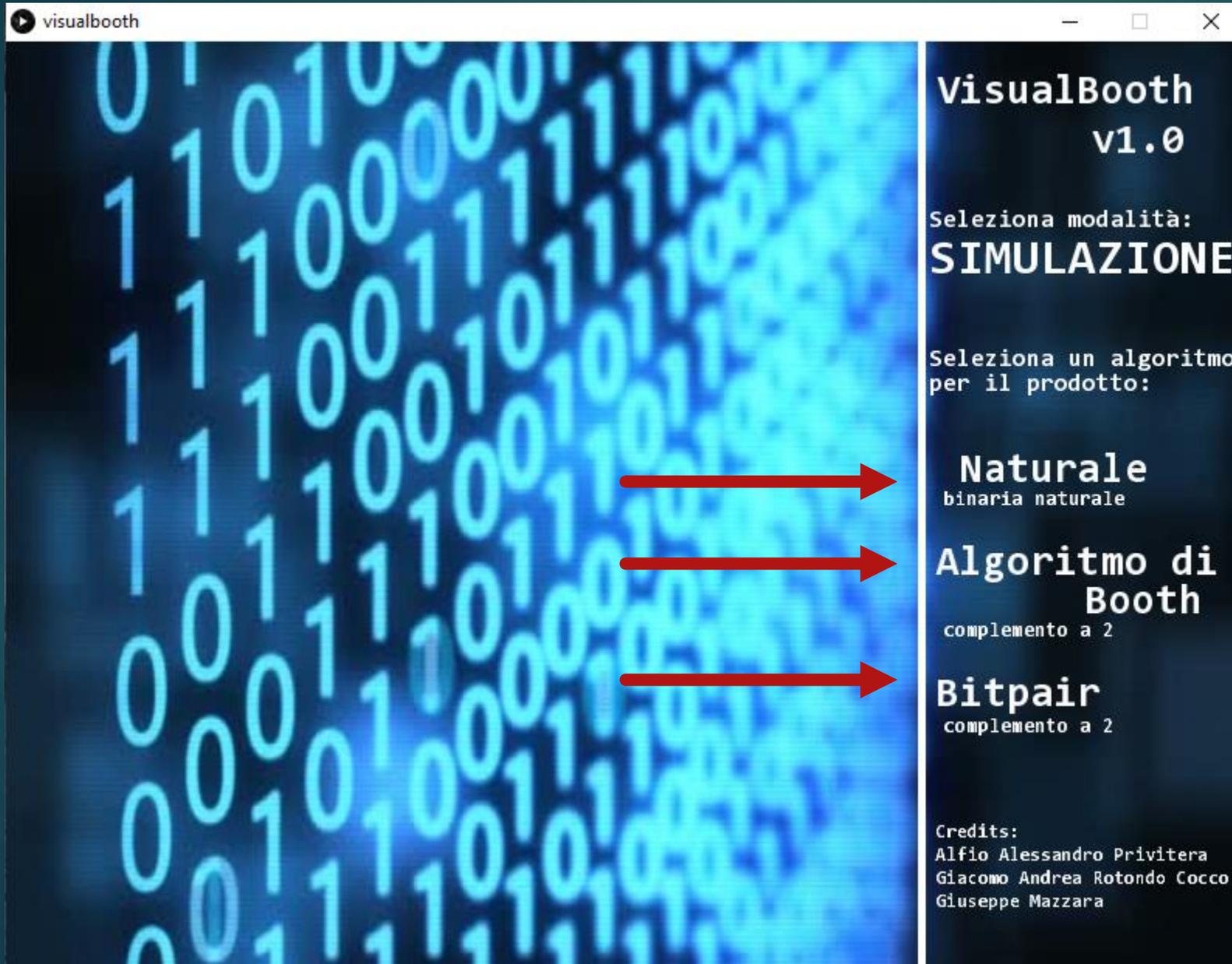
Algoritmo di Booth

- ▶ L'algoritmo di Booth, per il prodotto tra due cifre binarie in complemento a due, si divide in due fasi.
La prima fase prevede la conversione del secondo fattore con codifica di Booth:
- ▶ La codifica di Booth del secondo fattore si ricava nel seguente modo:
- ▶ Si aggiunge uno zero di supporto ai calcoli a destra della cifra meno significativa e si procede accoppiando la cifra presa in esame con quella alla sua destra, spostandosi di una cifra alla volta.
- ▶ La codifica per ogni coppia è la seguente: $11=0$; $00=0$; $10=-1$; $01=1$.
- ▶ La seconda fase si articola moltiplicando la codifica di Booth del secondo fattore ottenuta per il primo fattore. L'operazione si svolge in maniera analoga a quella in colonna, a differenza che quando la cifra è -1 si effettua il complemento a due del primo fattore come risultato del prodotto per la cifra della codifica

Algoritmo di Bitpair

- ▶ Esso è un'evoluzione dell'algoritmo di Booth, ideato per compiere meno operazioni, infatti funziona analogamente, con una codifica per il secondo fattore, basata a cascata su quella di Booth.
- ▶ Dopo aver ricavato la codifica di Booth per il secondo fattore, la codifica per il bitpair si ricava dalla codifica di Booth accoppiando a due a due le cifre a partire da quelle meno significative, spostandosi di due cifre alla volta.
- ▶ La codifica per le coppie è la seguente: $0b=b$, con $b \in \{-1, 0, +1\}$; $\underline{-1+1}=-1$; $\underline{-10}=-2$; $\underline{+10}=+2$; $\underline{+1-1}=+1$. Le coppie $\underline{+1+1}$ e $\underline{-1-1}$ non si presentano mai per la logica dell'algoritmo.
- ▶ Questa codifica porterà ad avere 4 cifre intervallate da zeri, i quali costituiscono delle operazioni nulle in cui si shifta il prodotto alla sinistra. Nel seguente Software questi sono rappresentati da spazi vuoti nella codifica.
- ▶ Una volta ottenuta la codifica la moltiplicazione si svolge in maniera analoga a quella con codifica di Booth, a differenza che quando la cifra della codifica è +2 si effettua uno shift di una posizione a sinistra della cifra, altrimenti se la cifra è -2, oltre allo shift a sinistra si fa la conversione del primo fattore in complemento a due.

Presentazione dell'applicativo



Avviando il Software il menù principale presenta tre opzioni per il prodotto, indicate in figura dalle frecce rosse.

► Esse sono:

- Prodotto naturale
- Algoritmo di Booth
- Prodotto con Bitpair

Prodotto naturale

visualbooth

Prodotto naturale
(rappresentazione binaria naturale)

START

0	0	1	0	1	0	1	0
0	1	0	1	1	0	1	0

X =

Ritorna al menù

PASSO 1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 2: 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0

PASSO 3: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 4: 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0

PASSO 5: 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0

PASSO 6: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 7: 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0

PASSO 8: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

RISULTATO: 0 0 0 0 1 1 1 0 1 1 0 0 0 1 0 0

Dall'istanza in immagine sarà possibile eseguire il prodotto come nell'usuale operazione sui numeri decimali, utilizzando numeri binari in rappresentazione binaria naturale. Per procedere sarà sufficiente alterare i valori contenuti nel moltiplicando e nel moltiplicatore in alto a destra cliccandoci sopra. Una volta impostati i numeri premendo START sarà possibile vedere ogni singolo passaggio fino al risultato.

Prodotto con algoritmo di Booth

visualbooth

Prodotto con algoritmo di Booth (rappresentazione in complemento a 2)

Ritorna al menù

START

cod. di Booth:

0	1	0	1	0	0	1	0	X
0	0	0	1	1	0	1	0	=
0	0	1	0	-1	1	-1	0	↷

PASSO 1:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 2:

1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 3:

0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 4:

1	1	1	1	1	1	0	1	0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 5:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 6:

0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 7:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 8:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

RISULTATO:

0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Dall'istanza in immagine sarà possibile eseguire il prodotto con algoritmo di Booth. Come per gli altri algoritmi sarà sufficiente alterare i valori contenuti nel moltiplicando e nel moltiplicatore in alto a destra cliccandoci sopra. Una volta impostati i numeri premendo START sarà possibile vedere ogni singolo passaggio fino al risultato.

Il software è inoltre in grado di mostrare la codifica di Booth del secondo fattore evidenziando al passaggio del puntatore le cifre prese in considerazione

Prodotto con Bitpair

visualbooth

Prodotto con Bitpair
(rappresentazione in complemento a 2)

Ritorna al menù

START

cod. BitPair: 1 -1 2 -2

	0	1	1	0	0	1	0	1	X
	0	0	1	1	0	1	1	0	=

PASSO 1: 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 0

PASSO 2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 3: 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0

PASSO 4: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 5: 1 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0

PASSO 6: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 7: 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0

PASSO 8: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

RISULTATO: 0 0 0 1 0 1 0 1 0 1 0 0 1 1 1 0

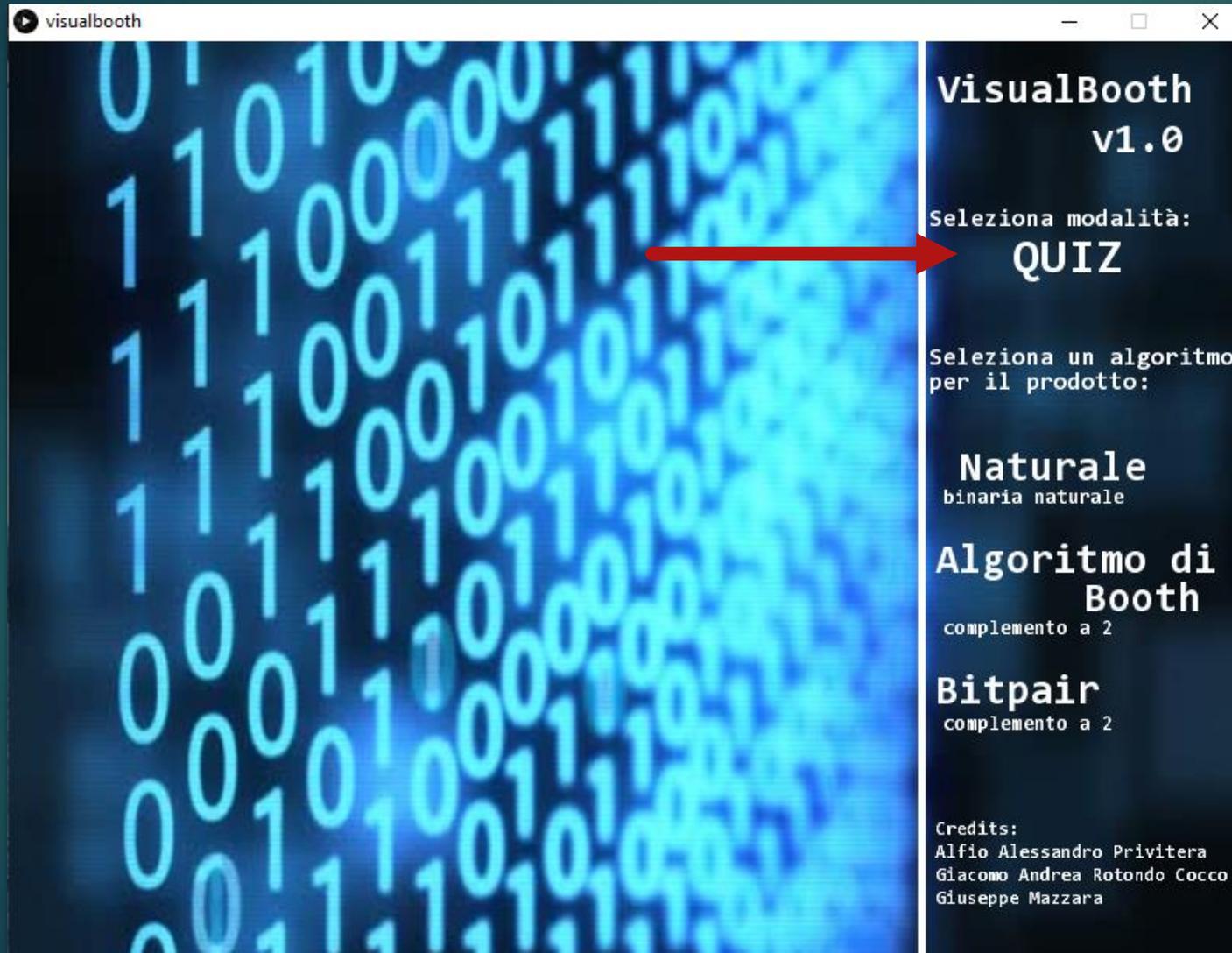
Analogamente a quanto visto per gli altri algoritmi è possibile eseguire il prodotto con bitpair alterando i valori contenuti nel moltiplicando e nel moltiplicatore in alto a destra cliccandoci sopra. Una volta impostati i numeri premendo START sarà possibile vedere ogni singolo passaggio fino al risultato.

Viene mostrata la codifica in BitPair del secondo fattore, la quale prevede quattro cifre alternate da operazioni nulle, in corrispondenza delle quali i passi dell'operazioni di moltiplicazione sono tutti zeri.

Quando la cifra della codifica è 2 o -2, una serie di «<<<» notifica che è avvenuto uno shift sinistro.

Per ognuno degli algoritmi presentati è possibile tornare indietro utilizzando la funzione apposita «Ritorna al menù» in alto a sinistra.

Modalità QUIZ



Per allenarsi e mettersi alla prova è stata implementata una modalità chiamata «quiz» (indicata dalla freccia rossa nell'immagine a sinistra) che permette di testare le proprie conoscenze in tutti e tre gli algoritmi: Prodotto Naturale, Algoritmo di Booth e Algoritmo Bitpair.

Una volta selezionata la modalità è sufficiente scegliere l'algoritmo desiderato.

Modalità QUIZ: Naturale

visualbooth

Prodotto naturale
(rappresentazione binaria naturale)

START

0 0 1 1 0 1 1 1 **X**
0 0 1 1 1 1 0 0 **=**

Ritorna al menù

PASSO 1: **0 0 0 0 0 0 0 0** **0 0 0 0 0 0 0 0**

PASSO 2: **0 0 0 0 0 0 0** **0 0 0 0 0 0 0 0**

PASSO 3: **0 0 0 0 0 0** **0 0 1 1 0 1 1 1** **0 0**

PASSO 4: **0 0 0 0 0** **0 0 1 1 0 1 1 1** **0 0 0**

PASSO 5: **0 0 0 0** **0 0 1 1 0 1 1 1** **0 0 0 0**

PASSO 6: **0 0 0** **0 0 1 1 0 1 1 1** **0 0 0 0 0**

PASSO 7: **0 0** **0 0 0 0 0 0 0 0** **0 0 0 0 0 0 0**

PASSO 8: **0** **0 0 0 0 0 0 0 0** **0 0 0 0 0 0 0 0**

RISULTATO: **0 0 0 0 1 1 0 0** **1 1 1 0 0 1 0 0** **OK**

QUIZ

DIFFICOLTA': FACILE.

In questa modalità nella sessione «prodotto naturale» l'obiettivo dell'utilizzatore è calcolare il risultato dei valori generati casualmente premendo il pulsante <start> in alto al centro.

Una comoda etichetta sulla destra ti ricorderà di essere nella modalità desiderata.

Se la soluzione è corretta il programma restituirà <OK> come nell'immagine di esempio a sinistra.

Modalità QUIZ: Algoritmo di Booth

visualbooth

Prodotto con
algoritmo di Booth (rappresentazione in complemento a 2)

Ritorna al menù

START

cod. di Booth:

1	0	0	0	1	1	1	1	X
1	0	1	0	0	1	1	1	=
-1	1	-1	0	1	0	0	-1	OK

PASSO 1:

0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 2:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 3:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 4:

1	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 5:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 6:

0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 7:

1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PASSO 8:

0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

RISULTATO:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

QUIZ

DIFFICOLTA': MEDIA.

In questa modalità nella sessione «Algoritmo di Booth» l'obiettivo dell'utilizzatore è indovinare prima la codifica di Booth e solo a quel punto calcolare il risultato.

Una prima etichetta <OK>, come in figura, appare non appena la codifica di Booth è stata indovinata.

Una prima etichetta <OK> appare alla destra della codifica di Booth non appena essa è stata indovinata.

Successivamente, se la soluzione è corretta, il programma restituirà <OK> in basso a destra.

Modalità QUIZ: Bitpair

visualbooth

Prodotto con **Bitpair**
(rappresentazione in complemento a 2)

Ritorna al menù

START

cod. BitPair: **1 1 -2 0** OK

PASSO 1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 3: 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0

PASSO 4: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +

PASSO 5: 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0

PASSO 6: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PASSO 7: 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0

PASSO 8: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

RISULTATO: 0 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 OK

QUIZ

DIFFICOLTA': DIFFICILE.

In questa modalità nella sessione «Bitpair» come per l'algoritmo di Booth l'obiettivo dell'utilizzatore è indovinare prima la codifica Bitpair e solo allora calcolare il risultato.

Questa è la modalità quiz più impegnativa delle tre, poiché bisognerà:

- prima calcolare separatamente la cod. di Booth, dalla quale poi ricavare quella Bitpair;

- oppure applicare la ricodifica Bitpair definita su otto possibili terne di bit (considerando anche il bit a destra della coppia).

Una prima etichetta <OK> appare alla destra della codifica di Bitpair non appena essa è stata indovinata. Successivamente, se la soluzione è corretta, il programma restituirà <OK> in basso a destra.

Riconoscimenti

- ▶ La versione è stata rilasciata in data 15/02/2020.
- ▶ La nascita e lo sviluppo del progetto sono da attribuire interamente a:
 - ▶ Alfio Alessandro Privitera
 - ▶ Giacomo Andrea Rotondo Cocco
 - ▶ Giuseppe Mazzara

Studenti dell'Università degli Studi di Catania, L-31 Informatica.

Grazie per l'attenzione.