Introduzione

Finora abbiamo visto come cifrare dei messaggi di testo ma la crittografia si può utilizzare per rendere illeggibili anche altri elementi, come ad esempio delle immagini. Per poterlo fare dobbiamo però differenziare le immagini in bianco e nero dalle immagini a colori.

Cifrare e decifrare immagini in bianco e nero.

Per chi non si ricordasse la lezione relativa alla rappresentazione delle immagini si ricorda che per i pixel bianchi usiamo un bit con valore 0 mentre per i pixel neri usiamo un bit con valore 1. Avremmo bisogno di una *trasformazione* per poter cifrare e decifrare le immagini:

0 Trasformazione dei pixel:

Pixel 1	Pixel 2	Pixel risultante
☐	☐	☐
☐	■	■
■	☐	■
■	■	☐

Questa *trasformazione* per funzionare ha bisogno di 2 pixel, che dovranno essere obbligatoriamente in bianco e nero. Al termine della *trasformazione* avremo un nuovo pixel che potrà essere bianco oppure nero.

Per capire di che colore dovrà essere dovremmo porci la seguente domanda: "il pixel 1 ed il pixel 2 sono entrambi dello stesso colore?"
- Se entrambi sono tutti bianchi o tutti neri allora il nuovo pixel sarà bianco.
- Se invece un pixel è nero e l'altro è bianco allora il nuovo pixel sarà nero.

Per i più grandi – Uno sguardo nel dettaglio

La trasformazione appena vista è in realtà una funzione chiamata XOR.
*La funzione XOR è molto semplice da usare: innanzitutto dobbiamo sapere che questa funzione lavora con i valori booleani che sarebbero i valori di verità **VERO** e **FALSO**. In informatica possiamo rappresentare i valori booleani utilizzando i bit.*
*Normalmente viene associato il valore **FALSO** al bit 0 ed il valore **VERO** al bit 1.*
Lo XOR per funzionare ha bisogno in input di due valori, dei bit e ci restituirà un bit che sarà 0 se i due valori passati sono uguali o 1 se i bit sono diversi:

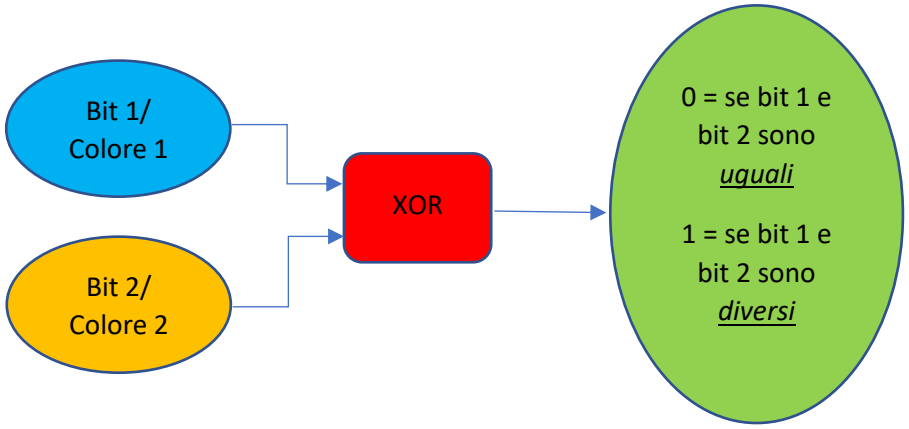


Tavola logica XOR

bit 1	bit 2	Risultato
0	0	0
0	1	1
1	0	1
1	1	0

Vi ricordate da cosa era composta la chiave che abbiamo utilizzato nel cifrario di Cesare? Avevamo utilizzato un numero che indicava il numero di spostamenti da effettuare all'interno dell'alfabeto.
Perché secondo voi per poter cifrare un'immagine è importante avere una trasformazione che prende in input due pixel? **Lasciare riflettere un attimo gli studenti e fateli intervenire.**

A differenza del cifrario di Cesare qui come chiave utilizzeremo un'altra immagine per effettuare la cifratura e la decifratura, anch'essa in bianco e nero.

Proviamo con un esempio:

Immagine A: da cifrare

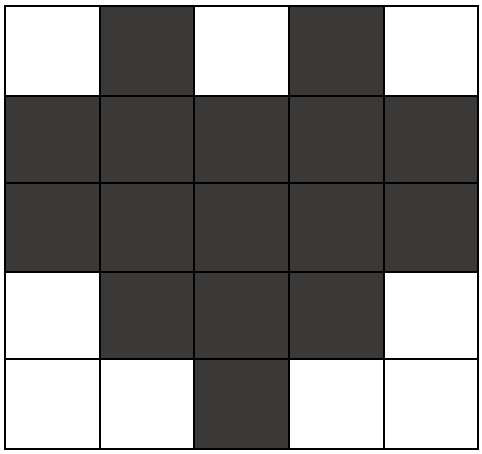
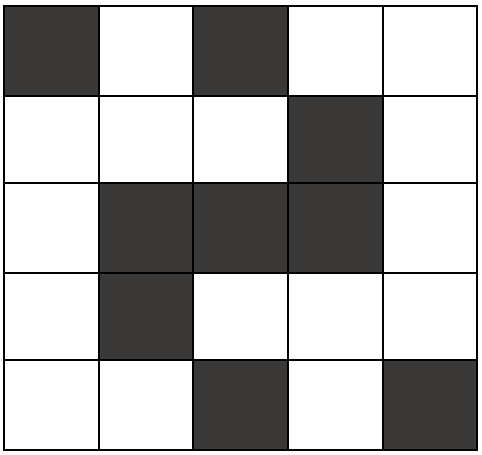


Immagine B: da usare come chiave



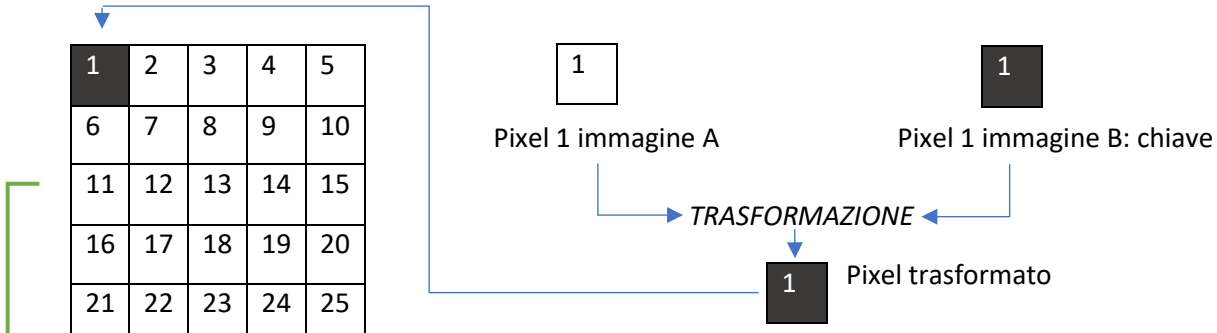
Per decifrare l'immagine dobbiamo applicare la trasformazione vista in precedenza prendendo i pixel delle immagini A e B che si trovano nella stessa posizione.
Per semplificarci il lavoro ed evitare di confonderci assegniamo dei numeri in ordine crescente ad ogni pixel, riga per riga:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

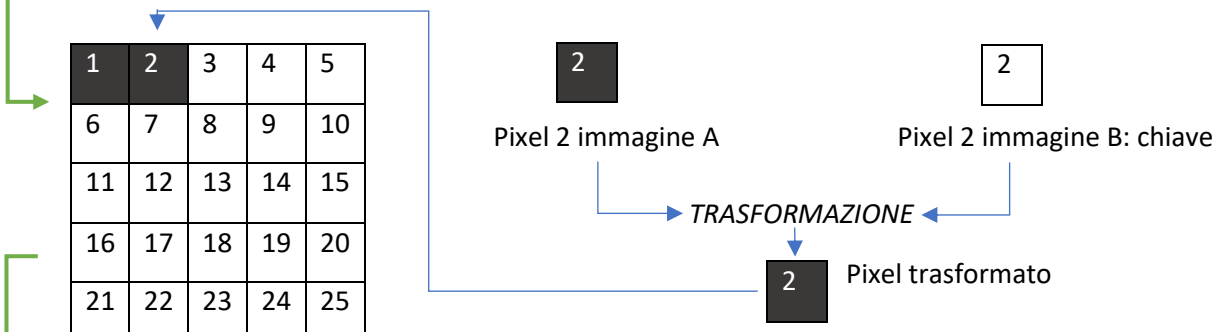
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Cifratura:

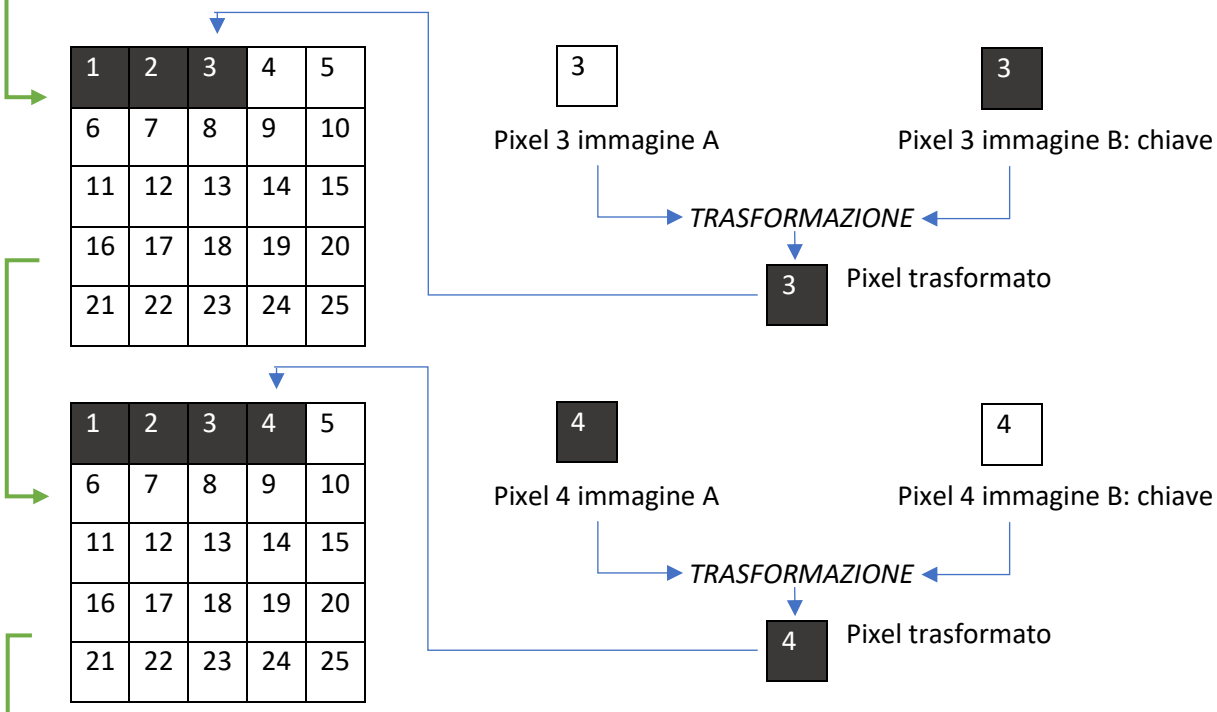
Creiamo ora un'immagine tutta bianca grande quanto l'immagine da cifrare e applichiamo la *trasformazione* per i pixel 1 e copiamo il risultato ottenuto al pixel corrispondente sull'immagine bianca:

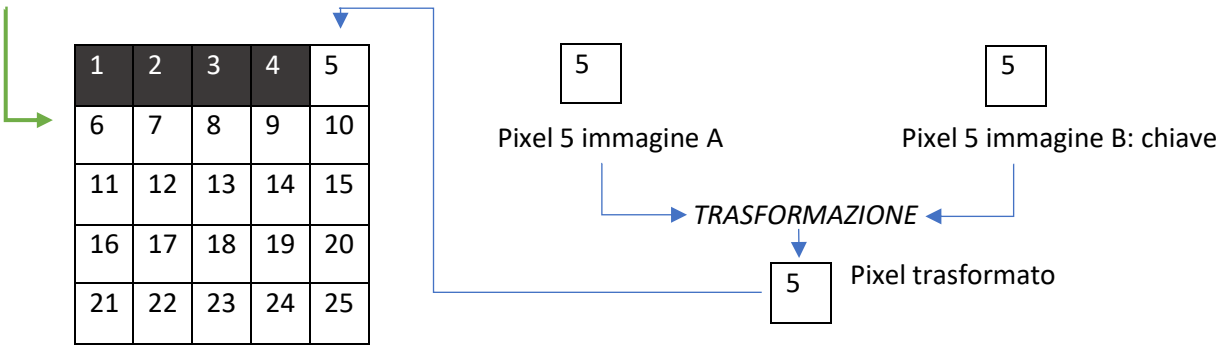


Successivamente riapplichiamo la stessa *trasformazione* ai pixel 2 andando a copiare il risultato ottenuto nell'immagine ottenuta prima:



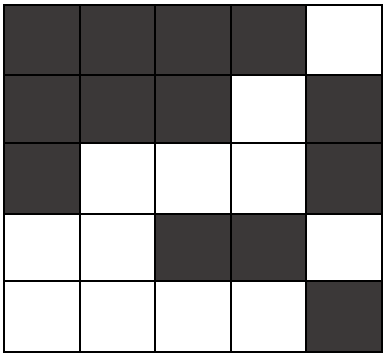
Ora rifacciamo la stessa operazione con i pixel 3, 4 e 5 in ordine:





Ripetendo questa operazione per tutti i pixel, quando avremo finito il lavoro troveremo una terza immagine, questa sarà l'immagine A cifrata:

Immagine A cifrata:

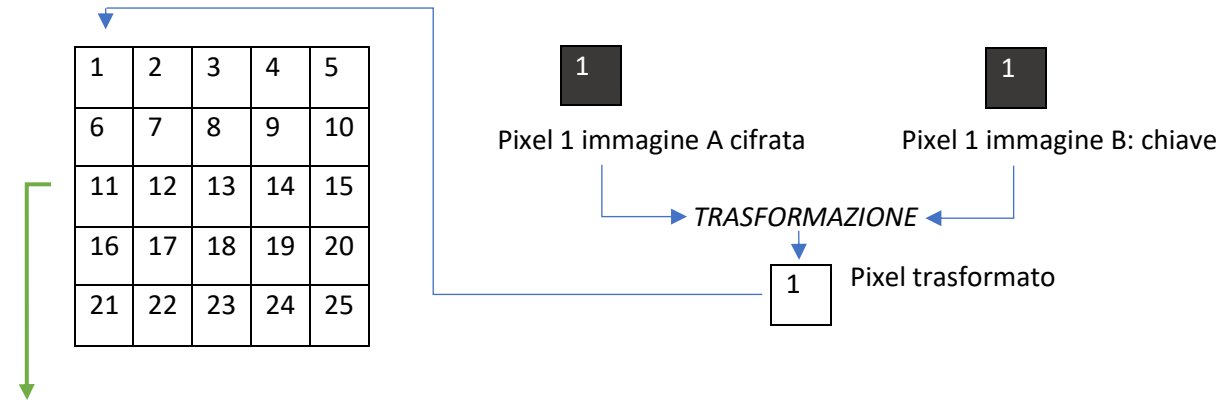


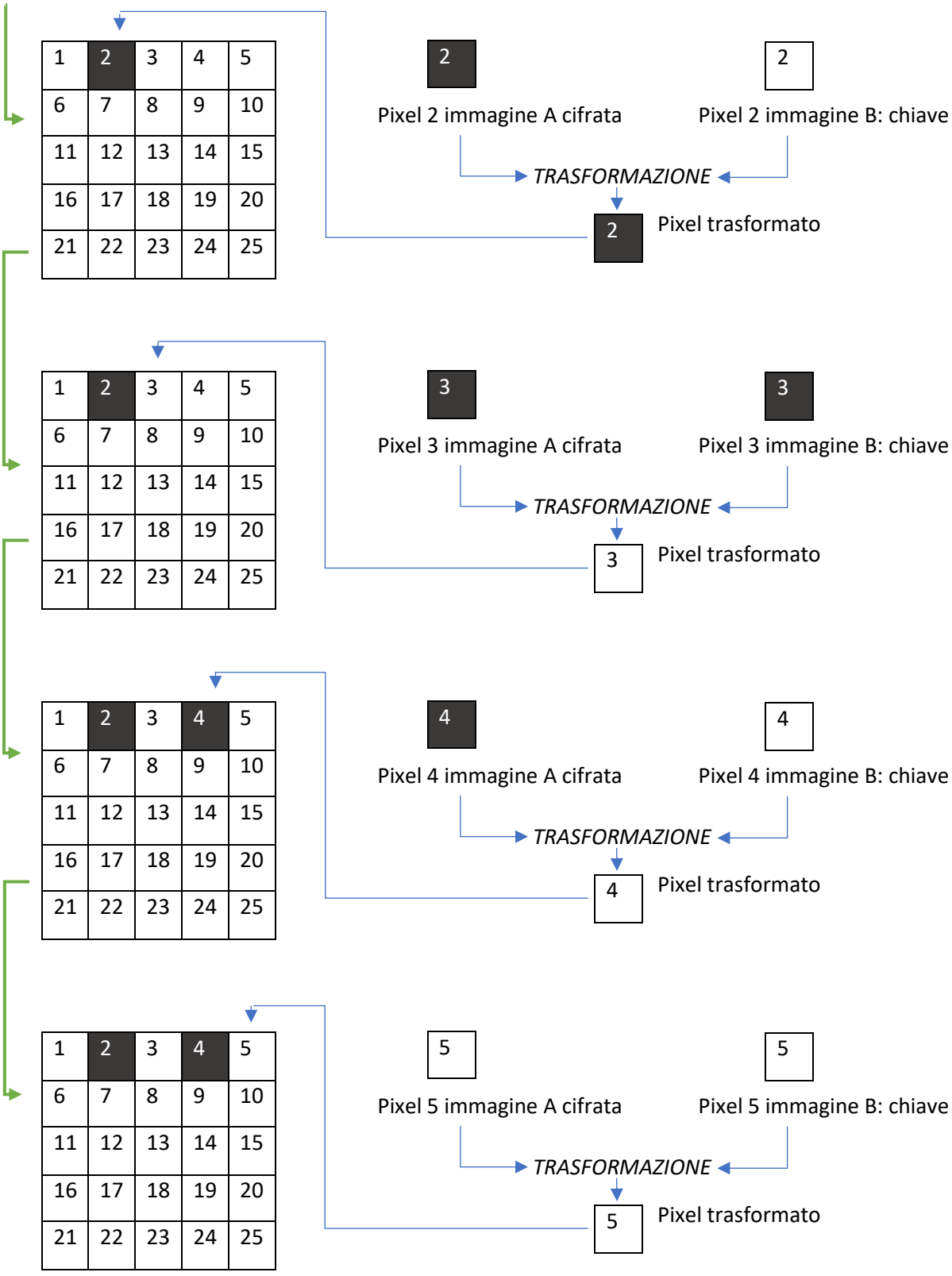
Questa immagine non sembra più un cuore! Come facciamo a ritornare indietro e a riavere il nostro bellissimo cuoricino? **Lasciare riflettere gli studenti e fateli intervenire. Se si trovano in difficoltà spostate la loro attenzione sulla tabella della Trasformazione con rappresentati i pixel e poi chiedetegli di provare a spiegare come hanno fatto a cifrare l'immagine.**

Decifratura:

Per decifrare l'immagine avremo sempre bisogno, oltre all'immagine cifrata, dell'immagine B, ovvero della *chiave.* È sufficiente poi ripetere le stesse operazioni fatte per cifrare l'immagine mediante la Trasformazione dei pixel su queste due immagini per riavere indietro il nostro cuore! L'immagine che decifreremo dovrà essere disegnata su una nuova immagine tutta bianca grande quanto l'immagine da decifrare.

Proviamo a decifrare i primi 5 pixel:




1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
2
Pixel 2 immagine A cifrata
2
Pixel 2 immagine B: chiave
TRASFORMAZIONE
2
Pixel trasformato
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
3
Pixel 3 immagine A cifrata
3
Pixel 3 immagine B: chiave
TRASFORMAZIONE
3
Pixel trasformato
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
4
Pixel 4 immagine A cifrata
4
Pixel 4 immagine B: chiave
TRASFORMAZIONE
4
Pixel trasformato
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
5
Pixel 5 immagine A cifrata
5
Pixel 5 immagine B: chiave
TRASFORMAZIONE
5
Pixel trasformato

Ripetendo questa operazione per tutti i pixel, quando avremo finito il lavoro troveremo una terza immagine, questa sarà l'immagine A decifrata:

Riecco il nostro cuore! 

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Esempio con ripetizione della chiave:

L'immagine che svolge il compito di chiave deve essere per forza grande quanto l'immagine da cifrare? No, ma è consigliato se si ha la possibilità perché se usiamo una chiave più piccola poi sarebbe un po' più facile riuscire ad intuire quale sia la chiave che abbiamo utilizzato!

Proviamo a cifrare questa immagine usando come chiave sempre l'immagine B, ripetendola tante volte, nel nostro caso 4, fino a costruire un'immagine della stessa dimensione dell'immagine D, un fungo:

Immagine D: da cifrare

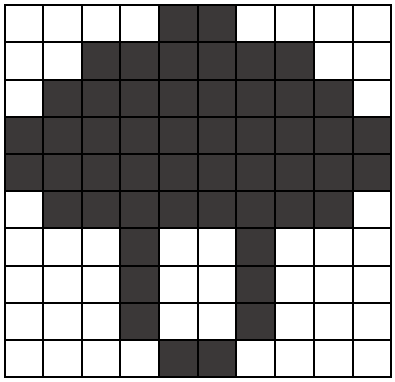
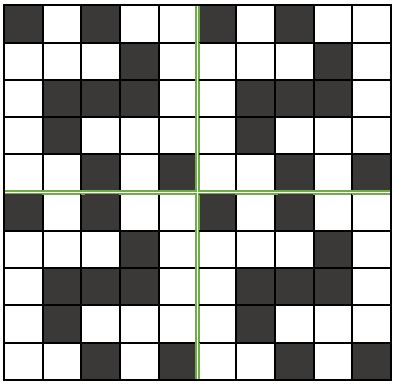


Immagine B: da usare come chiave



Suggerimento: se si ha difficoltà ad eseguire la cifratura perché le immagini sono troppo grandi dividere l'immagine D in quattro parti uguali come l'immagine B e procedere poi per righe per ogni blocco. Si consiglia inoltre di assegnare dei numeri ai pixel come fatto in precedenza, ma trattando ogni blocco come un'immagine a sé.

Blocco1

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 2

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 3

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 4

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Per cifrare queste 4 sotto immagini dobbiamo seguire lo stesso procedimento svolto nell'esempio precedente. Finita la cifratura dell'immagine D dovremmo trovarci con i seguenti blocchi cifrati:

Blocco 1 cifrato

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 2 cifrato

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 3 cifrato

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Blocco 4 cifrato

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Per decifrare questi quattro blocchi e ritornare all'immagine iniziale dobbiamo seguire lo stesso procedimento svolto nell'esempio precedente. Finita la decifratura dei quattro blocchi, eliminando gli spazi tra questi ritorneremo ad avere la nostra immagine iniziale:

Riecco il nostro fungo!