

Test 2

Consegna

Dovrai consegnare il codice sorgente e le classi compilate in una cartella nominata `Test<#>i2<Cognome><Nome>` (ad esempio `Test2i2BrambillaOttavio`).

Quando scrivi il codice sorgente assicurati di:

- indentarlo correttamente,
- commentarlo in modo appropriato,
- rispettare le convenzioni stilistiche di Java,
- rispettare rigorosamente le specifiche fornite,
- nominare correttamente i file.

SBRS Enconding

Il *Samt Bit Random Shift Encoding* (SBRSE) è un algoritmo di codifica utile per cifrare un file.

Per ogni *byte in chiaro* del *file sorgente*, il SBRSE produce una sequenza di 9 byte nel *file cifrato*: 1 *byte cifrato* e 8 *byte chiave*. Il *byte cifrato* contiene i bit del *byte sorgente* rimescolati in posizioni casuali specificate dai *byte chiave*. Ad ogni *byte in chiaro* viene applicata una differente *chiave* casuale.

I *byte chiave* (B1-B8) rappresentano le posizioni in cui i bit del *byte sorgente* si trovano dentro al *byte cifrato* (B0). P0 indica in quale posizione di B0 si trova il bit b0 del *byte in chiaro*, P1 indica la posizione del bit b1, P7 indica la posizione del bit b7, e via dicendo:

byte	B0	B1	B2	B3	B4	B5	B6	B7	B8
valore	VALORE	P0	P1	P2	P3	P4	P5	P6	P7

Ad esempio il *byte in chiaro*:

bit	b7	b6	b5	b4	b3	b2	b1	b0
valore	H	G	F	E	D	C	B	A

potrebbe essere codificato nel modo seguente:

byte	B0	B1	B2	B3	B4	B5	B6	B7	B8
valore	VALORE	1	7	0	4	6	5	2	3

il byte B0 (VALORE) conterrebbe:

bit	b7	b6	b5	b4	b3	b2	b1	b0
valore	B	E	F	D	H	G	A	C

Il byte VALORE conterrebbe i bit del *byte in chiaro* disposti nelle posizioni specificate da (B0-B7). Ovvero:

- il bit b0 (A) si trova nella posizione b1 di VALORE
- il bit b1 (B) si trova nella posizione b7 di VALORE
- il bit b2 (C) si trova nella posizione b0 di VALORE
- il bit b3 (D) si trova nella posizione b4 di VALORE
- il bit b4 (E) si trova nella posizione b6 di VALORE
- il bit b5 (F) si trova nella posizione b5 di VALORE
- il bit b6 (G) si trova nella posizione b2 di VALORE
- il bit b7 (H) si trova nella posizione b3 di VALORE

Cifrando tramite SBRSE un file di testo contenente 23 caratteri (ad esempio "Le fragole sono mature.") otterremmo un file cifrato di 207 byte composto da 23 sequenze di 9 byte.

Classe ByteDecipher [7pt.]

Realizza la classe `ByteDecipher` utile per decodificare il codice SBRSE. La classe dovrà esportare i seguenti metodi:

- `public static int getBit(int pos, byte value)`: ritorna il valore (0 o 1) del bit nella posizione `pos` nel byte `value`. Se la posizione specificata da `pos` non è valida ritorna 0.
- `public static byte decipherByte(byte[] encoded)`: riceve un array di 9 byte contenenti la codifica SBRSE di un byte e ritorna in valore decodificato.
- `public static byte[] decipherBytes(byte[] data)`: riceve un array contenente una sequenza di 9 X N byte codificati e ritorna un array contenente N byte decodificati.

Classe FileDecipher [6pt.]

L'applicativo `FileDecipher` serve per decifrare un file di testo cifrato in SBRSE. Riceve il percorso del file cifrato come argomento e ne stampa a terminale il contenuto decifrato:

- Se il file sorgente non esiste o se non è possibile leggerlo, stampa il messaggio "Lettura sorgente impossibile" e termina.
- Se durante la lettura del file avviene un errore, stampa il messaggio "Lettura sorgente fallita" e termina.
- Se il percorso del file sorgente non viene fornito, stampa il messaggio "Utilizzo: java FileDecipher <sorgente>" e termina.

Esempio:

```
> java FileDecipher segreto.sbrs
Le fragole sono mature.
>
```

Consegna

Implementa le classi `ByteDecipher` e `FileDecipher` rispettando le specifiche fornite. Dovrai consegnare una cartella nominata `Test2I2<Nome><Cognome>` contenente i codici sorgente e le classi compilate.