Esercizio

Il formato MOBI è utilizzato da diversi eBook reader, tra cui il famoso Kindle di Amazon. Il formato è una estensione del Palm Database Format (PDB) che consiste in un header di dimensione fissa, un numero variabile di informazioni sui record (Record Info Entry), e di seguito da altrettanti record. **Tutti i valori numerici sono codificati in Big Endian.**

L'header PDB per il formato MOBI ha la seguente struttura:

offset	bytes	content	comments
0	32	name	Nome del database (0 terminato)
32	2	attributes	(non importante)
34	2	version	file version
36	4	creation date	Numero di secondi dal 01/01/1904
40	4	modification date	Numero di secondi dal 01/01/1904
44	4	last backup date	Numero di secondi dal 01/01/1904
48	4	modificationNumber	(non importante)
52	4	appInfoID	(non importante)
56	4	sortInfoID	(non importante)
60	4	type	"BOOK"
64	4	creator	"MOBI"
68	4	uniqueIDseed	(non importante)
72	4	nextRecordListID	(non importante)
76	2	numberOfRecords	Numero di record

Seguono poi *numberOfRecords* Record Info Entry che hanno la seguente struttura:

bytes	content	comments
4	recordDataOffset	Offset nel file del record corrente
1	recordAttributes	(non importante)
3	uniqueID	Identificatore univoco del record

Il primo record (di solito con uniqueID = 0) contiene informazioni sull'eBook e in particolare inizia con un PalmDOC Header che ha la seguente struttura:

offset	bytes	content	comments
0	2	Compression	1 == no compression, 2 = PalmDOC compression, 17480 = HUFF/CDIC compression
2	2	Unused	(non importante)
4	4	TextLength	Lunghezza dell'intero testo non compresso.
8	2	RecordCount	Numero di PDB record utilizzati per il testo.
10	2	RecordSize	Dimensione massima del testo codificato in un record (sempre 4096)
12	2	EncryptionType	0 == no encryption, 1 = Old Mobipocket Encryption, 2 = Mobipocket Encryption
14	2	Unknown	(non importante)

In questo record si trovano poi ulteriori dati che non considereremo.

Dopo questo record, i successivi *RecordCount* record contengono il testo compresso con una variante dell'LZ77. In particolare per ogni byte del record compresso si verifica il suo valore e si compiono diverse azioni in base a questo:

<table>
<tr><th>valore (hex)</th><th>azione</th></tr>
<tr><td>00</td><td>si interrompe la decodifica del record corrente</td></tr>
<tr><td>01-08</td><td>si copiano in output i successivi 1-8 byte</td></tr>
<tr><td>09-7F</td><td>si manda in output il byte così com'è.</td></tr>
<tr><td>80-BF</td><td>si applica la decodifica in stile LZ77. Il byte corrente (80-BF) e il successivo vengono considerati un'unica sequenza di 16 bit e decodificata come:

<table>
<tr><td colspan="8">primo byte</td><td colspan="8">secondo byte</td></tr>
<tr><td>1</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td colspan="2">ignora</td><td colspan="11">distanza</td><td colspan="4">lunghezza-3</td></tr>
</table>

1) I due bit più significativi del primo byte 80-BF sono sempre 10 e vengono ignorati.
2) Gli altri 6 assieme ai 5 più significativi del byte successivo (11 in totale) formano un intero da 0 a 2047, con il valore 0 che non deve mai verificarsi.
3) Gli ultimi 3 del byte successivo forniscono un valore da 0 a 7 che va incrementato di 3 e quindi una lunghezza da 3 a 10.

Bisogna quindi spostarsi indietro di distanza (1-2047) byte nei dati decompressi finora e copiare in output lunghezza (3-10) byte a partire da quella posizione.</td></tr>
<tr><td>C0-FF</td><td>si applica una codifica per le coppie spazio+carattere: si manda in output uno spazio seguito dal byte corrente (C0-FF) a cui viene azzerato il bit più significativo (ad esempio con uno XOR con 0x80 o un AND 0x7F).</td></tr>
</table>

Scrivere il programma a linea di comando `MOBIdecode` in grado di estrarre da file in questo formato il testo non compresso. Il software deve supportare la seguente sintassi:

```
MOBIdecode <input filename> <output filename>
```

I due argomenti sono rispettivamente il nome del file da decodificare e quello da produrre in output. Tutti gli argomenti sono obbligatori.

Si seguano i seguenti step intermedi nel preparare la soluzione (assicurarsi che nel passare ad uno step successivo il precedente non venga distrutto ed è obbligatorio che quello che consegnate compili correttamente):
1) analizzare la linea di comando, aprire il file di input, creare il file di output, scriverci dentro il Byte Order Mark (BOM) per l'UTF-8 ovvero i 3 byte EF BB BF , leggere l'header PDB e stampare in output:

```
PDB name: <nome del database>↲
Creation date (s): <creation date>↲
Type: <type>↲
Creator: <creator>↲
Records: <numberOfRecords>↲
↲
```

2) leggere tutte le Record Info Entry e stampare in output (dopo le stampe precedenti):

```
0 - offset: <recordDataOffset del record 0> - id: <uniqueID del record 0>↲
1 - offset: <recordDataOffset del record 1> - id: <uniqueID del record 1>↲
…
↲
```

3) posizionarsi all'offset del primo record, leggere il PalmDOC Header e stampare in output (dopo le stampe precedenti):

```
Compression: <Compression>↲
TextLength: <TextLength>↲
RecordCount: <RecordCount>↲
RecordSize: <RecordSize>↲
EncryptionType: <EncryptionType>↲
↲
```

4) posizionarsi all'offset del record successivo, decodificarlo con l'algoritmo descritto in precedenza fino ad ottenere 4096 byte di output. A quel punto il primo record è concluso e **bisogna ignorare qualsiasi byte successivo**. Salvare la decodifica del primo record sul file di output (ovviamente dopo il BOM scritto in precedenza). Il file di output è un HTML con qualche tag particolare, ma se aperto con un browser qualsiasi dovrebbe vedersi correttamente senza simboli strani.

5) concludere l'esercizio ripetendo la decodifica sui successivi *RecordCount* record, considerando che un record è concluso quando da quel record sono stati ottenuti 4096 byte. Per quanto riguarda l'ultimo record (dei *RecordCount* dopo il primo) interrompere la decodifica quando si sono raggiunti i *TextLength* byte complessivi di output. Dopo questi record, ce ne sono altri che contengono indici, segnalibri e immagini, ma vanno ignorati.

Attenzione al Big Endian - Consegnate solo codice che compila - Non distruggete quanto fatto fino ad un certo step per passare al successivo (copia incolla dell'intera soluzione).