

Regionális forduló

2016. november 19.

9-10. osztályosok feladata

Feladat

Írjatok Markdown → HTML konvertert!

A markdown egy nagyon népszerű, nyílt forráskódú projektekben gyakran használt, jól olvasható jelölőnyelv, amely egyszerű ASCII karakterekkel jelzi többek között a *dőlt* és **félkövér** szöveget, a címeket vagy hiperhivatkozásokat.

Példa:

input.md

```
# Címsor 1

Ez itt egy *kiemelés*. Ez pedig annak a példája,
hogyan lehet fontos szöveget írni.

## Címsor 2

Ha szeretnénk, könnyen lehet [linkeket is](http://pelda.hu) írni
markdownban.
```

Ha ezt nyomtatásban tekintenénk meg, akkor a következőt látnánk:

Címsor 1Ez itt egy *kiemelés*. Ez pedig annak a példája, hogyan lehet **fontos szöveget** írni.**Címsor 2**Ha szeretnénk, könnyen lehet [linkeket is](#) írni markdownban.

A fenti markdown esetében a létrehozandó HTML-fájl a következő:

output.html

```
<html>
<body>
<h1>Címsor 1</h1>
<p>
    Ez itt egy <em>kiemelés</em>. Ez pedig annak a példája, hogy
    hogyan lehet <strong>fontos szöveget</strong> írni.
</p>
<h2>Címsor 2</h2>
<p>
    Ha szeretnénk, könnyen lehet <a href="http://pelda.hu">linkeket
    is</a> írni markdownban.
</p>
</body>
</html>
```

A konverternek be kell olvasnia az aktuális könyvtárban található *input.md* állományt, el kell végeznie a konverziót, és az eredményt az *output.html* fájlba kell kiírnia.

Markdown jelölések és azok HTML-megfelelői

A megvalósítandó konverternek nem kell a valóságban előforduló összes markdown jelölést támogatnia. Valójában ez lehetetlen feladat is lenne, hiszen a markdown nem egy szabványos jelölőnyelv, és így sok párhuzamos változata létezik, amelyek különbözőképpen valósítják meg az eredeti koncepciót. Az elkészítendő konverter programnak is egy speciális jelrendszert kell értelmeznie.

A markdown fájl html megfelelőjének minden esetben tartalmaznia kell a szabványos html és body tageket¹ (lásd fenti példa). A tartalomnak a body tagen belül kell elhelyezkednie. A html fájlt tetszőlegesen formázhatjuk, a tageket pl. kezdhetjük új sorban a könnyebb olvashatóság miatt, de az is lehetséges, hogy a teljes html fájl akár egy hosszú sorba kerül.

A markdown fájlban az egymás utáni sorok egy bekezdést alkotnak, amennyiben nincs közöttük üres sor. Ilyenkor az egyes sorok között tetszőleges whitespace² karaktert lehet használni elválasztásra.

Minden markdown fájlhoz érvényes html fájlt kell tudni generálni, ami azt jelenti, hogy a html fájlokban lévő tageknek mindig párban, lezárva kell lenniük, azaz szabályos struktúrát kell alkot-

¹ A tag a HTML kódban egy egységet jelölő elem. A <html> nyitó és </html> záró tag-ek a html kódot nyitják meg, majd zárják le.

² A whitespace karakterek ASCII kódjai: tab (9); kocsivissza (13); soremelés (10); szóköz (32)

niuk. Nem érvényes html pl. a következő fájl, mert nincs lezárva a <p> tag, a tag és az tag pedig egymást keresztezik:

```
<html>
<body>
  <p>Ez nem <em>ervenyes <strong> xhtml. </em> </strong>.
</body>
</html>
```

A konverternek a következő markdown jelöléseket szükséges támogatnia:

Az elem szerepe, jelentése	Markdown jelölés	HTML megfelelő
Bekezdés	A bekezdések egybefüggő nem üres sorok. Két bekezdést tetszőleges számú üres sor választhat el. Megjegyzés: Üres sor minden olyan sor, amely a whitespace karaktereken kívül más karaktert nem tartalmaz.	<p>bekezdés szövege</p>
Kiemelés	*kiemelendő szöveg* Megjegyzés: Feltételezhetitek, hogy a kiemelés a bekezdésen belül mindig le van zárva.	kiemelendő szöveg
Fontos szöveg	**fontos szöveg** Megjegyzés 1: Feltételezhetitek, hogy a fontos szöveg a bekezdésen belül mindig le van zárva. Megjegyzés 2: Feltételezhetitek, hogy egy szövegrész sem lehet egyszerre kiemelő szöveg és fontos szöveg is. Megjegyzés 3: Feltételezhetitek, hogy a markdownban nem szerepel ***	fontos szöveg
Link	[link szövege] (http://pelda.hu) Megjegyzés 1: Feltételezhetitek, hogy a linken belül a markdownban nem szerepel kiemelés vagy fontos szöveg. Fontos szövegben vagy kiemelésben sem szerepelhet link.	link szövege

	<p>Megjegyzés 2: Ha az url rész tartalmaz " karaktert, azt helyettesíteni kell a %22 sztringgel a html-kimenetben.</p> <p>Megjegyzés 3: Feltételezhetitek, hogy a link mindig helyesen szerepel a markdownban, azaz nincs lezáratlan zárójel, azok megfelelő sorrendben követik egymást, és a] után mindig (következik whitespace nélkül.</p> <p>Megjegyzés 4: Ha a markdownban szerepel egy [,], (vagy) karakter, az csakis link céljából fordul elő.</p>	
Kisebb jel (<)	<	<
Nagyobb jel (>)	>	>
Címsor 1	Az a bekezdés, amely sor elején a # karakterrel, majd utána egy szóközzel kezdődik.	<pre><h1> szöveg </h1></pre>
Címsor 2	Az a bekezdés, amely a sor elején a ## karakterekkel, majd utána egy szóközzel kezdődik.	<pre><h2> szöveg </h2></pre>
Címsor 3 Címsor 4 Címsor 5 Címsor 6	<p>A Címsor 1 és Címsor 2 mintájára...</p> <p>Megjegyzés: Feltételezhetitek, hogy a # karakter csak a címsor megjelölése céljából és helyes szintaktikával fordulhat elő a szövegben.</p>	A Címsor 1 és Címsor 2 mintájára...

Beadandó

A program **forráskódja** (a programozási környezettől függően a forráskód több fájl is lehet, esetleg mappák is tartozhatnak hozzá), és a **lefordított fájl *verseny*** fájlneven, ha a programozási környezet a fordítást támogatja.

A program által használt külső fájlokat mindig az aktuális könyvtárba kell elhelyezni (A program ne tartalmazzon abszolút elérési útvonalat!)

Kódolási alapelvek

A forráskód minőségét is értékeljük.

Irányelvek, szempontok:

- Egységes kódolási szabályok az azonosítókra:
 - a változók egységes elnevezése (kis- és nagybetűk vagy más speciális karakterek használata),

- az osztályok egységes elnevezése (objektum-orientált programnyelv esetén),
- a függvények és eljárások tartalomra utaló elnevezése,
- a programkód egységes strukturáltsága, tagoltsága.
- A kód minősége (könnyen – emberek számára – érhető illetve karbantartható kód):
 - áttekinthető, lehetőség szerint rövid eljárások, függvények, fájlok,
 - beszédes, tömör elnevezésű azonosítók,
 - objektum-orientált nyelveknél globális változók mellőzése.
- Kommentezés:
 - A kommentezés elsődleges célja, hogy a programban a miért? kérdésre adjon választ. (A mit? kérdésre az azonosítók megfelelő elnevezése és a megfelelően strukturált kód, a hogyan? kérdésre pedig az áttekinthető forráskód ad választ.)
 - A túlzásba vitt kommentezés csökkenti az áttekinthetőséget, a túl kevés komment nehezíti a megértést.
 - Elvárás a változók, osztályok, függvények és eljárások szerepének rövid, értelem-szerű kommentezése.

Elérhető pontszám: 100 pont (Ebből a helyes dokumentálás – kommentezés – 15 pont)

Jó munkát kíván a versenybizottság!