Zadání úlohy do projektu z předmětu IPP 2015/2016
(Obecné a společné pokyny všech úloh jsou v `proj2016.pdf`)

SYN: Zvýraznění syntaxe

Zodpovědný cvičící: Jiří Kučera (ikucera@fit.vutbr.cz)

1 Detailní zadání úlohy

Vytvořte skript pro automatické zvýrazňování různých částí textu. Skript bude pracovat s tabulkou regulárních výrazů, ke kterým bude přiřazeno požadované výstupní formátování.

Tento skript bude pracovat s těmito parametry:

- `--help` viz společné zadání všech úloh
- `--format=filename` určení formátovacího souboru. Soubor bude obsahovat libovolné množství formátovacích záznamů. Formátovací záznam se bude skládat z regulárního výrazu vymezujícího formátovaný text a příkazů pro formátování tohoto textu. Detailní popis viz níže.
- `--input=filename` určení vstupního souboru v kódování UTF-8.
- `--output=filename` určení výstupního souboru opět v kódování UTF-8 s naformátovaným vstupním textem. Formátování bude na výstupu realizováno některými HTML elementy na popis formátování. Regulární výrazy ve formátovacím souboru určí, který text se naformátuje, a formátovací informace přiřazené k danému výrazu určí, jakým způsobem se tento text naformátuje.
- `--br` přidá element `<br />` na konec každého řádku původního vstupního textu (až po aplikaci formátovacích příkazů).

1.1 Formátovací soubor

Formátovací soubor se bude skládat z jednotlivých řádků (formátovacích záznamů) v následujícím formátu:

<regulární výraz>`\t+`*<seznam formátovacích příkazů>*

tedy každý neprázdný řádek bude obsahovat *<regulární výraz>* oddělený jedním nebo více tabulátory od *<seznamu formátovacích příkazů>*. Seznam příkazů bude tvořen jednotlivými příkazy oddělenými čárkou a libovolným počtem mezer nebo tabulátorů. Na každou část textu, která vyhovuje danému regulárnímu výrazu, se budou aplikovat HTML elementy určené následujícím seznamem formátovacích příkazů. Následují jednotlivé příkazy, jejich případný stručný popisek a příslušný formátovací HTML kód:

bold tučný text: `<b>text</b>`

italic kurzíva: `<i>text</i>`

underline podtrhnutí: `<u>text</u>`

teletype `<tt>text</tt>`

size:[**cislo**] velikost textu, kde `[cislo]` představuje číslo od 1 do 7:
`<font size=[cislo]>text</font>`

color:[**hex**] barva textu, kde `[hex]` znázorňuje hexadecimální číslo od `000000` do `FFFFFF`:
`<font color=#[hex]>text</font>`

Jednotlivé HTML elementy se aplikují do výsledného textu v takovém pořadí, v jakém jsou napsané ve formátovacím souboru a v rámci jednotlivých formátovacích záznamů (v seznamu formátovacích příkazů záleží na pořadí zleva doprava).

Příklad formátovácího souboru, vstupu a výstupu:
Formátovací záznam:

```
aa*    italic, color:FF0000
```

Vstupní text:	Výstupní text:
`Toto je priklad bbbaaabbb`	`Toto je prikl<i><font color=#FF0000>a</font></i>d  bbb<i><font color=#FF0000>aaa</font></i>bbb `

1.2 Regulární výrazy

Regulární výrazy budou definovány podobně jako na přednáškách předmětu IFJ. Všechny znaky s ASCII hodnotou 32 a vyšší budou představovat regulární výraz popisující řetězec složený jen z daných symbolů (s výjimkou speciálních symbolů `.|!*+()%`, které budou použité pro tvorbu složitějších výrazů). Další regulární výrazy jsou definovány rekurzivně následujícím způsobem:

Nechť `A` a `B` jsou regulární výrazy. Potom:

- `A.B` - popisuje řetězce vyhovující výrazu `A` následované libovolným řetězcem vyhovujícím výrazu `B`.
- `AB` - zkratka pro `A.B`.
- `A|B` - popisuje řetězce vyhovující buď `A`, nebo `B`.
- `!A` - negace výrazu `A`. Všechny takové řetězce, které nevyhovují výrazu `A`. Negace je aplikovatelná pouze na výrazy popisující jeden znak nebo na speciální výrazy tvaru `%X` popsané níže.
- `A*` - libovolné opakování řetězců vyhovujících výrazu `A` (i nulový počet opakování).
- `A+` - nenulový počet opakování řetězců výrazu `A` (tj. `A.A*`).
- `(A)` - závorky pro určení priority.

Priorita jednotlivých operátorů (od nejvyšší k nižší): `!` > `*`, `+` > `.` > `|`.
Speciální regulární výrazy:

- `%s` - bílé znaky (`\t\n\r\f\v`)
- `%a` - jeden libovolný znak
- `%d` - čísla od 0 do 9

- `%l` - malá písmena od `a` do `z`
- `%L` - velká písmena od `A` do `Z`
- `%w` - malá a velká písmena, tj. `(%l|%L)`
- `%W` - všechna písmena a čísla, tj. `(%w|%d)`
- `%t` - znak tabulátoru (`\t`)
- `%n` - znak nového řádku (`\n`)
- `%`<*speciální symbol*> - znak <*speciální symbol*> (dostupné symboly: `.|!*+()%`)

Příklad:

Řádek formátovacího souboru		Význam
`if|then|else`	`bold`	klíčová slova tučně
`"(!"*)"`	`italic,color:FFFF44`	řetězcový literál, kurzíva + žlutá barva
`(100|%d%d|%d)%%`	`size:4, underline`	procenta (tj. `0%` až `100%`) velké písmo + podtrhnutí

1.3 Poznámky a speciální případy

- V případech, že nebyl zadán formátovací soubor, nebo nejde otevřít, nebo je prázdný, se toto nepovažuje za chybu a výstupem bude vstupní soubor.
- Ve výsledném souboru formátujte pouze neprázdné řetězce (i když regulárnímu výrazu ve formátovacím záznamu může příslušet i prázdný řetězec).
- V případě parametru `--br` přidejte element `<br />` na konec řádku (tj. před symbol `\n`) až po aplikování všech formátovacích pravidel.
- V případě nejednoznačnosti se při hledání dle zadaného regulárního výrazu hledá vyhovující řetězec maximální délky.
- Výrazu `!%a` nevyhovuje žádný znak (ani prázdný řetězec).

Následující příklad ilustruje aplikaci formátování při překrývání jednotlivých regulárních výrazů:

Formátovací soubor:

```
Takto se resi     bold
resi prekryvani   italic
```

Vstupní text:	Výstupní text:
`Takto se resi prekryvani`	`<b>Takto se <i>resi</b> prekryvani</i>`

Jak je vidět z příkladu, na přidávané formátovací elementy se regulární výrazy nevztahují.

Reference:

- MEDUNA, Alexander a LUKÁŠ, Roman. *Přednášky předmětu Formální jazyky a překladače (IFJ): Kapitola III. Modely regulárních jazyků*. FIT VUT v Brně, 2011. [cit. 2014-02-09]. Dostupné z: `https://www.fit.vutbr.cz/study/courses/IFJ/private/prednesy/Ifj03-cz.pdf`
- Regular expression. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2012-01-28 [cit. 2012-02-09]. Dostupné z: `http://en.wikipedia.org/wiki/Regular_expression`

2 Poznámky k hodnocení:

Hodnocení bude automatizované pomocí nástroje `diff`, proto prosím věnujte zvýšené úsilí formálním požadavkům na formát výsledného souboru (pořadí a syntaxe jednotlivých HTML elementů, formátování jenom neprázdných řetězců, ukončení řádku pomocí `<br />` v případě přepínače `--br` apod.).

3 Bonusová rozšíření

- **HTM** (HTML validita, 1 bod): Rozšíření se skládá ze dvou funkcionalit — zamezení překrývání tagů a escapování HTML tagů ve vstupním souboru, které budou spuštěny pomocí dvou parametrů `--nooverlap` a `--escape`.

  1) Zamezení překrývání tagů (parametr `--nooverlap`) bude probíhat na základě tohoto pravidla: pokud je tag otevřen uvnitř jiného tagu, musí být uvnitř stejného tagu také uzavřen (jak je tomu u HTML). Tato definice ale umožňuje více možností řešení, proto bude správně vyhodnocena pouze následující varianta: nejdříve se vloží všechny otevírací tagy, a až v případě, že by uzavřením nějakého tagu došlo k překrytí, tak se blokující tag rozdělí na více částí. Tato technika je demonstrována na následujícím příkladu:

     Příklad. Mějme formátovací soubor

     ```
     3+        bold
     (2|3)+    italic
     (1|2)+    underline
     ```

     a na vstupu `111222333`. Dle požadavků se nejprve aplikuje `bold`, potom `italic` a nakonec `underline`. Jako meziprodukt tedy postupně dostaneme (vložíme-li nejdříve otevírací tagy)

     ```
     111222<b>333          # 1. krok - aplikuji bold
     111<i>222<b>333       # 2. krok - aplikuji italic
     <u>111<i>222<b>333    # 3. krok - aplikuji underline
     ```

     a finální správný výstup potom postupně bude

     ```
     <u>111<i>222<b>333                    # 1. krok - uzavření underline:
                                           #    dojde k překrytí <i>, tedy:
     <u>111<i>222</i><b>333                #    a. uzavři <i>
     <u>111<i>222</i></u><b>333            #    b. uzavři <u>
     <u>111<i>222</i></u><b><i>333         #    c. otevři <i> za <b>
     ```

```
<u>111<i>222</i></u><b><i>333</i>      # 2. krok - uzavřu italic
<u>111<i>222</i></u><b><i>333</i></b>  # 3. krok - uzavřu bold
```

Poznámky: Pozor na prioritu otevíracích tagů mezi překryvy (`<b><i>` v příkladu) — ta se řídí původním zadáním, tzn. odpovídá prioritě dané formátovacím souborem. Bez parametrů musí program splňovat základní funkčnost popsanou v zadání.

2) Escapování HTML tagů ve vstupním souboru (parametr `--escape`), tzn. nahradit znaky `<`, `>` a `&` za `&lt;`, `&gt;` a `&amp;`.

- **NQS** (1 bod): Správné vyhodnocení regulárních výrazů obsahujících více kvantifikátorů za sebou. Správné vyhodnocení by mělo plynout z definice regulárního výrazu v zadání. Pro názornost pár příkladů (`A` je regulární výraz):

```
A++ = A+
A** = A*
A+* = A*
A*+ = A*
```