

DOMÁCÍ ÚKOL, SÉRIE Č. 4
10. LISTOPADU 2016

1. **Úloha** (4 body). Uvažujte následující situaci. Máte soubor, ve kterém je na každém řádku uveden jeden student UK spolu s relevantními daty (t.j. jméno, příjmení, rok nastoupení, fakulta). Napište program, který z tohoto souboru vytvoří nový soubor, který bude obsahovat loginy všech studentů z původního souboru.

Login studenta je tvořen prvními čtyřmi znaky jeho příjmení, prvním znakem jeho jména, poslední číslicí roku, ve kterém nastoupil na fakultu, jedním rozlišovacím písmenem a prvním písmenem názvu fakulty. Rozlišovací písmeno je standardně 'a'. Pokud už jsou dva studenti, kteří by jinak měli stejný login, první má rozlišovací písmeno 'a', druhý 'b'. Podobně pro tři studenty.

Příklad

Příjmení:	Jarník
Jméno:	Vojtěch
Rok nástupu:	1915
Fakulta:	Přírodovědecká fakulta
Login:	jarnv5ap

Presný formát vstupního a výstupního souboru si stanovte sami. Správným použitím rozlišovacích písmen zajistíte, že žádní dva studenti nebudou mít stejný login.

2. **Úloha** (4 body). Napište program, který se uživatele zeptá na název souboru a číslo n . Pak zadaný soubor načte a vypíše n nejčastěji se vyskytujících písmen anglické abecedy (ostatní znaky ignorujte) v daném souboru. (Funkce `sorted` uspořádá seznam čísel/řetězců podle velikosti/abecedy; všimněte si, že pro čísla mezi $[0, 1]$ je uspořádání podle abecedy stejné jako podle velikosti)

3. **Úloha** (2 body). Napište program, který načte český text ze souboru, odstraní z něj háčky a čárky (t.j. ve výsledném souboru bude např. místo znaku 'á' znak 'a') a zapíše ho do nového souboru. Můžete předpokládat, že vstupní soubor obsahuje pouze české znaky, mezery a znaky pro konec řádky.

4. **Úloha** (2 body). Napište program, který načte ze souboru posloupnost čísel (na každém řádku bude jedno číslo) a pak vypíše největší číslo, nejmenší číslo, průměr posloupnosti a směrodatnou odchylku.

5. **Úloha** (1 bod). Bez použití `if/else` napište program, který načte číslo ze vstupu a pokud bude větší, než 3.143214, pak vypíše "ANO" jinak vypíše "NE". (Hint: použijte funkci `atan`.)

6. **Úloha** (4 body). Napište program, který načte textový soubor a vypíše seznam slov, které se v něm vyskytují. U každého slova navíc vypíše počet výskytů.

Následující šifrovací postup je připisován Juliu Caesarovi. Chtějí-li si Alice s Bobem předat šifrovanou zprávu, předem se dohodnou na číslo $0 < n < 26$. Zpráva se zašifruje tak, že každé písmeno zprávy se nahradí písmenem, které se v abecedě nachází o n písmen dále. Příklad ($n = 3$)

Zpráva	S	E	J	D	E	M	E	S	E	V	D	E	V	E	T
Šifra	V	H	M	G	H	P	H	V	H	Y	G	H	Y	H	W

Suetonius, životopisce Julia Caesara, tvrdil, že Caesar tuto šifru používal pro komunikaci se svými generály. Je velmi pravděpodobné, že v Caesarově době to byl naprosto bezpečný způsob komunikace, zejména proto, že většina jeho protivníků

byla negramotná. Pro nás je však tento šifrovací systém velmi slabý, ačkoliv stojí za zmínku, že ještě na začátku 20. století byl používán Carskou armádou. Její protivníci však ve své většině již negramotní nebyli. . .

7. Úloha (3 body). Napište program, který přečte na vstupu anglickou zprávu zašifrovanou pomocí Caesarovy šifry a pokusí se tuto zprávu dešifrovat. K odhadnutí posunutí n využijte toho, že delší zpráva v angličtině bude zcela jistě obsahovat člen 'the'.

E-mail address: `jonathan.verner@ff.cuni.cz`