

Meno:	Tibor Dulovec	Hodnotenie projektu: (max 10(TS)/5(RAM) bodov)
Cvičenie:	Pondelok 9:00, J. Vincúr	
Dátum:	30. 4. 2021	

Projekt TZIV LS2020/21 – TS

Zadanie:	Na vstupe sú dátumy narodenia ľudí v tvare <i>dd.mm.yyyy%<pohlavie></i> oddelené pomlčkami. V prípade, ak sa jedná o muža, má <i><pohlavie></i> hodnotu 0, v prípade ženy 1. Vstup je z množiny znakov: cifry (0-9), bodka (‘.’), pomlčka (‘-’) a znak %. Navrhnete Turingov stroj - riešenie, ktoré pre každý zadáný dátum vypíše na výstupe prvých 6 čísl rodneho čísla osoby a informáciu či je šesťčísle párne alebo nie oddelené mriežkou – pre párne vypíše 0 pre nepárne vypíše 1, výstupy pre jednotlivé dátumy oddelí pomlčkou.
Vstup:	Akceptované vstupy: 20.03.1986%0-20.03.1986%1\$, 31.02.1986%1-01.12.2000%1\$, 20.03.1986%0-20.03.1986%1\$, 01.12.2000%1\$, 20.03.0986%0 Neakceptované vstupy: 99.03.1986%0-20.13.1986%1\$, 20.03.9999%0-20.03.1986%1\$, 20.03.1999%7\$, 20.83.1986%1\$, 20.13.1986%\$
Neformálne riešenie:	Prvým ktorom je presunúť sa na začiatok, keďže prvotná pozícia môže byť kdekoľvek na páske. Následne sa prepisujú cifry sprava doľava po páske. Začne sa rokom, kde sa na koniec pásky prepíšu iba posledné dve cifry. Potom sa hlava posunie na mesiace a následne na dni. Podľa toho, či je percento podčiarknuté, nadčiarknuté alebo oboje hlava vie, ktoré čísla má prepisovať. Pri prepisovaní prvej cifry mesiaca, sa ešte skontroluje číslo za percentom a podľa toho sa rozhodne, či sa na výstupe vypíše číslo o 5 väčšie ako je prvá cifra mesiaca alebo nie. Po prepísaní všetkých dni sa vloží na koniec pásky # a zaň 0 alebo 1, podľa toho, či posledná cifra vytvoreného rodneho čísla je párna. Po skončení prepísania jedného záznamu, tak sa prechádzanie po ďalšom zázname končí od pomlčky po Blank na konci pásky. Ak už na páske nie je žiadny iný záznam, prejde sa každý prvok na páske zľava doprava a každý upravený prvok sa prepíše na pôvodný pred upravením.

Zložitosť riešenia:	Vzhľadom nato, že jeden záznam vstupu, je vždy rovnako veľký, tak jeho náročnosť bude vždy podobná. Celková zložitosť vykonania následne rastie podľa počtu záznamov rozdelených pomlčkou na vstupe.
Simulátor:	Sim Studio
<p>Definícia výpočtového modelu (prechodová funkcia), kód simulátora (copy-paste):</p> <pre> ALL_TO_END = {0,1,2,3,4,5,6,7,8,9, 0_,1_,2_,3_,4_,5_,6_,7_,8_,9_, ., %, #, -, \$, ._, %_, -_, __.} ALL_FOR_# = {0,1,2,3,4,5,6,7,8,9, 0_,1_,2_,3_,4_,5_,6_,7_,8_,9_, ., %, #, -, \$, ._, %_, -_, __.} GROUP_DASH = {0,1,2,3,4,5,6,7,8,9, ., %, #, \$} CISLA = {0,1,2,3,4,5,6,7,8,9} podCISLA = {0_,1_,2_,3_,4_,5_,6_,7_,8_,9_} EVEN = {0,2,4,6,8} ODD = {1,3,5,7,9} OK = {0,1,2,3,4,5,6,7,8,9,-,%,.,#,,\$} // Prechody na vloženie vsetkych cisiel a znakov f(insertDash, GROUP_DASH) = (insertDash, GROUP_DASH, R) f(insertDash, -) = (insertDash, -_, R) f(insertDash, Blank) = (start, -, L) f(insert#0, _) = (insert#0, __, R) f(insert#0, ALL_FOR_#) = (insert#0, ALL_FOR_#, R) f(insert#0, Blank) = (insert0, #, R) f(insert#1, _) = (insert#1, __, R) f(insert#1, ALL_FOR_#) = (insert#1, ALL_FOR_#, R) f(insert#1, Blank) = (insert1, #, R) f(insert0, ALL_TO_END) = (insert0, ALL_TO_END, R) f(insert0, Blank) = (start, 0, L) f(insert1, ALL_TO_END) = (insert1, ALL_TO_END, R) f(insert1, Blank) = (start, 1, L) f(insert2, ALL_TO_END) = (insert2, ALL_TO_END, R) f(insert2, Blank) = (start, 2, L) f(insert3, ALL_TO_END) = (insert3, ALL_TO_END, R) f(insert3, Blank) = (start, 3, L) f(insert4, ALL_TO_END) = (insert4, ALL_TO_END, R) f(insert4, Blank) = (start, 4, L) f(insert5, ALL_TO_END) = (insert5, ALL_TO_END, R) f(insert5, Blank) = (start, 5, L) f(insert6, ALL_TO_END) = (insert6, ALL_TO_END, R) f(insert6, Blank) = (start, 6, L) f(insert8, ALL_TO_END) = (insert8, ALL_TO_END, R) f(insert8, Blank) = (start, 8, L) </pre>	

```

f(insert9, ALL_TO_END) = (insert9, ALL_TO_END, R)
f(insert9, Blank) = (start, 9, L)

// Posunutie na zaciatok pasky pri spusteni programu
f(q0, ALL_TO_END) = (start, ALL_TO_END, L)
f(start, ALL_TO_END) = (start, ALL_TO_END, L)
f(start, Blank) = (findRecord, Blank, R)

// Prechod na dalsi zaznam prvku alebo vlozeni pomlcky po dokončení záznamu
f(start, _%) = (insertDash, _%, R)
f(start, -_) = (findRecord, -_, R)

// Prechádzanie záznamom po percento, ktoré symbolizuje stav prepisovania a
podľa toho vyberie ďalšiu akciu
f(findRecord, CISLA) = (findRecord, CISLA, R)
f(findRecord, podCISLA) = (findRecord, podCISLA, R)
f(findRecord, #) = (findRecord, #, R)
f(findRecord, .) = (findRecord, ., R)
f(findRecord, ._) = (findRecord, ._, R)
f(findRecord, _.) = (findRecord, _., R)
// vybrat a prepísať rok
f(findRecord, %) = (findYear, %, L)
// vybrat a prepísať mesiac
f(findRecord, %_) = (findMonth, _%, L)
// vybrat a prepísať den
f(findRecord, _%) = (findDay, _%, L)

// Posun na zaciatok roka a prepísanie 3 a 4tej cifry
f(findYear, CISLA) = (findYear, CISLA, L)
f(findYear, .) = (skipMillennium, ._, R)
f(findYear, podCISLA) = (theYear, podCISLA, R)

// Preskocenie tisicrocia a storocia
// Ak je tisicrocie väčšie ako 2, nastane chyba
f(skipMillennium, 0) = (skipCentury, 0_, R)
f(skipMillennium, 1) = (skipCentury, 1_, R)
f(skipMillennium, 2) = (skipCentury, 2_, R)

f(skipCentury, 9) = (decade, 9_, R)
f(skipCentury, 8) = (decade, 8_, R)
f(skipCentury, 7) = (decade, 7_, R)
f(skipCentury, 6) = (decade, 6_, R)
f(skipCentury, 5) = (decade, 5_, R)
f(skipCentury, 4) = (decade, 4_, R)
f(skipCentury, 3) = (decade, 3_, R)
f(skipCentury, 2) = (decade, 2_, R)
f(skipCentury, 1) = (decade, 1_, R)
f(skipCentury, 0) = (decade, 0_, R)

// 3ta cifra roka
f(decade, 9) = (insert9, 9_, R)
f(decade, 8) = (insert8, 8_, R)
f(decade, 7) = (insert7, 7_, R)
f(decade, 6) = (insert6, 6_, R)
f(decade, 5) = (insert5, 5_, R)
f(decade, 4) = (insert4, 4_, R)
f(decade, 3) = (insert3, 3_, R)
f(decade, 2) = (insert2, 2_, R)
f(decade, 1) = (insert1, 1_, R)

```

```

f(decade, 0) = (insert0, 0_, R)

// 4ta cifra roka
f(theYear, %) = (findMonth, %_, L)
f(theYear, 9) = (insert9, 9_, R)
f(theYear, 8) = (insert8, 8_, R)
f(theYear, 7) = (insert7, 7_, R)
f(theYear, 6) = (insert6, 6_, R)
f(theYear, 5) = (insert5, 5_, R)
f(theYear, 4) = (insert4, 4_, R)
f(theYear, 3) = (insert3, 3_, R)
f(theYear, 2) = (insert2, 2_, R)
f(theYear, 1) = (insert1, 1_, R)
f(theYear, 0) = (insert0, 0_, R)

// Posun na zaciatok mesiaca
f(findMonth, podCISLA) = (findMonth, podCISLA, L)
f(findMonth, CISLA) = (findMonth, CISLA, L)
f(findMonth, ._) = (findMonth, _., L)
f(findMonth, .) = (1month, ., R)
f(findMonth, _) = (2month, ._, L)

// ak prva cifra mesiaca je 0, treba skontrolovat cislo za % a podla toho
vypisat na pásku 0 alebo 1
f(1month, 0) = (1month0, 0_, R)
f(1month0, CISLA) = (1month0, CISLA, R)
f(1month0, podCISLA) = (1month0, podCISLA, R)
f(1month0, ._) = (1month0, ._, R)
f(1month0, _) = (1month0, _., R)
f(1month0, %_) = (1month0c, %_, R)
f(1month0c, 0) = (insert0, 0_, R)
f(1month0c, 1) = (insert5, 1_, R)

// ak prva cifra mesiaca je 1, treba skontrolovat cislo za % a podla toho
vypisat na pásku 0 alebo 1
f(1month, 1) = (1month1, 1_, R)
f(1month1, CISLA) = (1month1, CISLA, R)
f(1month1, podCISLA) = (1month1, podCISLA, R)
f(1month1, ._) = (1month1, ._, R)
f(1month1, _) = (1month1, _., R)
f(1month1, %_) = (1month1c, %_, R)
f(1month1c, 0) = (insert1, 0_, R)
f(1month1c, 1) = (insert6, 1_, R)

// vypis druhej cifra mesiaca
f(2month, 9) = (insert9, 9_, R)
f(2month, 8) = (insert8, 8_, R)
f(2month, 7) = (insert7, 7_, R)
f(2month, 6) = (insert6, 6_, R)
f(2month, 5) = (insert5, 5_, R)
f(2month, 4) = (insert4, 4_, R)
f(2month, 3) = (insert3, 3_, R)
f(2month, 2) = (insert2, 2_, R)
f(2month, 1) = (insert1, 1_, R)
f(2month, 0) = (insert0, 0_, R)

// Posun na koniec dna v zázname
f(findDay, podCISLA) = (findDay, podCISLA, L)
f(findDay, ._) = (findDay, _., L)
f(findDay, .) = (findDay, _., L)
f(findDay, CISLA) = (day1, CISLA, L)

```

```

f(findDay, _) = (day2, _, L)

// vypis prvej cifry dna (max 3)
f(day1, 3) = (insert3, 3_, R)
f(day1, 2) = (insert2, 2_, R)
f(day1, 1) = (insert1, 1_, R)
f(day1, 0) = (insert0, 0_, R)

// vypis druhej cifry dna
f(day2, 9) = (insert9, 9, R)
f(day2, 8) = (insert8, 8, R)
f(day2, 7) = (insert7, 7, R)
f(day2, 6) = (insert6, 6, R)
f(day2, 5) = (insert5, 5, R)
f(day2, 4) = (insert4, 4, R)
f(day2, 3) = (insert3, 3, R)
f(day2, 2) = (insert2, 2, R)
f(day2, 1) = (insert1, 1, R)
f(day2, 0) = (insert0, 0, R)

// Ked je pridany cely datum, prida sa hashtag a cislo podla toho, ci je
posledna cifra parna alebo neparna
f(day1, podCISLA) = (checkDay2, podCISLA, R)
f(checkDay2, EVEN) = (insert#0, EVEN, R)
f(checkDay2, ODD) = (insert#1, ODD, R)

// Ked nieje ziadny dalsi zaznam, na paske sa vratia upravene cisla do
povodneho stavu
f(findRecord, -) = (checkOneRecordDash, Blank, L)
f(checkOneRecordDash, -) = (clear, Blank, L) // pre vymazanie pomlcky za
jednym zaznamom
f(checkOneRecordDash, CISLA) = (clear, CISLA, L)
f(clear, OK) = (clear, OK, L)
f(clear, -) = (clear, -, L)
f(clear, .) = (clear, ., L)
f(clear, %) = (clear, %, L)
f(clear, 0_) = (clear, 0, L)
f(clear, 1_) = (clear, 1, L)
f(clear, 2_) = (clear, 2, L)
f(clear, 3_) = (clear, 3, L)
f(clear, 4_) = (clear, 4, L)
f(clear, 5_) = (clear, 5, L)
f(clear, 6_) = (clear, 6, L)
f(clear, 7_) = (clear, 7, L)
f(clear, 8_) = (clear, 8, L)
f(clear, 9_) = (clear, 9, L)
// Ukoncenie programu po prichode na zaciatok pasky
f(clear, Blank) = (qf, Blank, R)

```