

KSI 2012

Úloha 1-2: Poetická

Jan Horáček
Gymnázium, Brno, Vídeňská 47; jan.horacek@seznam.cz

21. září 2012

1 Úvod

Zadání této úlohy si kladlo důležitou podmínku, tedy, že v celém řešení se nesmí vyskytovat ani jeden cyklus. Zpočátku se zdá takové řešení poměrně složité, ale po uvědomění si, že cyklus lze relativně snadno nahradit rekurzí, je problém hezky řešitelný.

2 Zdrojový kód

Zdrojový kód k tomuto problému je přiložen v souboru *Reseni.txt* v kódování UTF-8 a byl vytvořen <http://ksi.fi.muni.cz/karel/> osobně.

3 Popis řešení

Program se spouští funkcí *NASOB*.

Program je založen na definici funkce *COPY*, která rekurzivně zkopíruje počet značek z jednoho pole do druhého. Tato funkce je pak opakovaně volána a tím je docíleno násobení.

Následuje kompetní slovní popis funkce programu.

3.1 Funkce *COPY*

Funkce *COPY* je založena na rekurzivním opakování sbírání značek a následném umísťování střídavě na cílové a na výchozí pole.

Budiž pole definované jako *pos* takové pole, na kterém se Karel nachází při vstupu do této funkce.

Funkce kontroluje počet značek na poli *pos*. Pokud je počet značek větší, než 0, funkce zvedne značku a zavolá sama sebe. Jakmile na poli *pos* není žádná značka, začne se funkce z rekurzivně vytvořeného zanoření vracet zpět na původní úroveň. Při tomto vynořování dochází k umísťování značek střídavě na pole *pos* (pro zjištění počtu značek jsme je museli sebrat) a na pole *pos + 1*.

Výsledkem funkce je tedy pole *pos + 1*, ve kterém se nachází počet značek stejný, jako v poli *pos*.

3.2 Funkce *NASOB*

Funkce *NASOB* zapouzdřuje funkci *COPY* do jakéhosi cyklu, ale tento cyklus je implementován jako rekurzivní volání sama sebe.

Budiž pole definované jako *pos* takové pole, na kterém se Karel nachází při započetí programu.

Funkce zvedá z daného pole *pos* značky. Po zvednutí jedné značky dojde ke zkopírování počtu prvků z pole $pos + 1$ do pole $pos + 2$, což zajistí funkce *COPY*. Záměrně zde uvádím *pos* namísto klasického *x*, nebo *y*, protože program může fungovat ve všech 4 směrech. Směr násobení definuje výchozí směr robota Karla.

Celý algoritmus popsany v přechozím odstavci se opakuje, dokud je na poli *pos* značka.

4 Závěr

Program je principálně funkční.

Pevně doufám, že neexistuje žádná časově-nenáročná funkce na zjištění počtu značek na poli, protože pokud by existovala, byl by můj algoritmus značně časově neefektivní.

Reference

Tento algoritmus byl vytvořen pouze autorovou vlastní silou - bez použití externích zdrojů.