

Recunoaștere de texte

19 octombrie 2010

Extragerea numerelor de telefon dintr-un text

Un text contine, printre altele, numere de telefon scrise in formatul `(prefix)numar`. Afişați lista numerelor de telefon din text.

Exemplu:

Dragi parinti, ieri mi-am cumparat o cartela Orange de pe care o sa va pot suna ieftin seara dupa ora 17. Am prins o oferta buna (3 dolari) si am si SMS-uri gratuite (1000). Numarul nou este (0744)343434. Sa stiti ca si numarul vechi a ramas valabil (0723)564323. Ion.

Se va afișa:

0744-343434

0723-564323

O primă soluție

La intrare putem avea **cifră**, (,) sau **altceva** (orice caracter diferit de cele de mai sus). În plus poate să apară simbolul **EOF** (sfârșit de fișier).

Sistemul se poate afla în orice moment în una din stările

- ▶ **T**: text oarecare
- ▶ **P**: prefix al unui număr de telefon
- ▶ **N**: număr de telefon
- ▶ **-**: prelucrarea s-a încheiat

Starea inițială este **T**.

Regulile de tranziție

	cifră	()	altceva	EOF
T	T	P	T	T	-
P	P	T	N	T	-
N	N	T	T	T	-

Cod C (prima versiune)

Trei funcții recursive:

- ▶ `text`
- ▶ `prefix`
- ▶ `numar`

```
void prefix(void);
```

```
void numar(void);
```

```
void text(void)
```

```
{  
    int c = getchar();  
    // daca am '(' trec la prefix  
    if (c == '(') prefix();  
    // altfel raman la text  
    else if (c != EOF) text();  
}
```

Cod C (prima versiune)

```
void prefix(void)
{
    int c = getchar();
    // cat timp am cifre , raman la prefix
    if (isdigit(c)) {
        putchar(c);
        prefix();
    }
    // prefixul e gata , trec la numar
    else if (c == ')') {
        putchar('-');
        numar();
    }
    // eroare , prefixul nu ar trebui sa se incheie altfel
    else if (c != EOF) {
        putchar('\r'); // sterg linia curenta
        text();
    }
}
```

Cod C (prima versiune)

```
void numar(void)
{
    int c = getchar();
    // cat timp am cifra , raman la numar
    if (isdigit(c)) {
        putchar(c);
        numar();
    }
    // cand nu mai am cifra revin la text
    else if (c != EOF) {
        putchar('\n'); // inchei linia curenta
        text();
    }
}

int main(void)
{
    text(); // initial sunt in starea T
    return 0;
}
```

Problemă cu soluția prezentată

Programul prezentat acceptă și un prefix sau număr vid:

1000-

0744-343434

0723-564323

Trebuie să numărăm câte cifre au apărut în prefix și în număr.

Cod C (a doua versiune)

```
void prefix(int n);  
void numar(int n);  
  
void text(void)  
{  
    int c = getchar();  
    // daca am '(' trec la prefix  
    if (c == '(') prefix(0);  
    // altfel raman la text  
    else if (c != EOF) text();  
}
```


Cod C (a doua versiune)

```
void prefix(int n)
{
    int c = getchar();
    // cat timp am cifre , raman la prefix
    if (isdigit(c)) {
        putchar(c);
        prefix(n + 1);
    }
    // prefixul e gata , trec la numar
    // daca am exact 4 cifre in prefix
    else if (c == ')' && n == 4) {
        putchar('-');
        numar(0);
    }
    // eroare , prefixul nu ar trebui sa se incheie altfel
    else if (c != EOF) {
        putchar('\r'); // sterg linia curenta
        text();
    }
}
```

Cod C (a doua versiune)

```
void numar(int n)
{
    int c = getchar();
    // cat timp am cifra , raman la numar
    if (isdigit(c)) {
        putchar(c);
        numar(n + 1);
    }
    // cand nu mai am cifra revin la text
    else if (c != EOF) {
        // daca am exact 6 cifre in numar, inchei linia
        // altfel sterg linia (numar invalid)
        if (n == 6) putchar('\n');
        else putchar('\r');
        text();
    }
}

int main(void) {
    text(); // initial sunt in starea T
    return 0;
}
```

Evaluare de expresii

Se dă o expresie aritmetică formată din:

- ▶ operatori: $+$ $-$ $*$ $/$
- ▶ operanzi: numere întregi
- ▶ paranteze: $()$

Scrieți un program care evaluează expresia și afișează rezultatul.
Operatorii au precedența uzuală.

Exemplu

$20 * (5 + 4 * (2 / 6)) * (10 / 2 + 12 - 8) / 5$

Se va afișa

228

Soluție

Putem identifica reguli de construcție a expresiei.

$$E \rightarrow T$$

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$T \rightarrow F$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$F \rightarrow \text{număr}$$

$$F \rightarrow (E)$$

Trei funcții recursive: `expr`, `termen` și `factor`.

Atenție la asociativitate și la spațiile albe.