Cum citim de la intrare

– un *cuvânt* (orice până la spaţiu alb)

```
char s[80]; scanf("%79s", s);
```

spaţiu alb = spaţiu sau `\f \n \r \t \v`
ignoră spaţii albe iniţiale; adaugă `'\0'` la sfârşit
Atenţie! Nu se poate citi *o linie* de text în acest fel !
din `Ana are mere` va citi doar primul cuvânt: `Ana`

– o *linie de text*, până la `'\n'`

```
char s[80]; fgets(s, 80, stdin);
```

citeşte max. 80-1 caractere, inclusiv `'\n'`, adaugă `'\0'` la sfârşit

`stdin`: identificator definit în `stdio.h` pt. fişierul standard de intrare

Cum testăm sfârşitul de fişier

1. În orice punct de program: `int feof(FILE *fp)`
returnează nenul (adevărat) dacă s-a atins sfârşitul lui fp; 0 dacă nu
pentru fişierul de intrare: `feof(stdin)`

2. După valoarea returnată de funcţiile de intrare:

```
int c; c = getchar(); if (c == EOF) /* ...  */
```

Atenţie! `c` trebuie declarat `int` pentru a testa de `EOF`
valoarea `EOF` (-1) e diferită de cea a oricărui caracter (`0 .. 255`)

`scanf` returnează `EOF` (-1) dacă întâlneşte imediat sfârşitul de fişier
(nu şi dacă a reuşit să citească măcar ceva)

⇒ Folosiţi doar `if (scanf(...) == nr_variabile_dorite)` pentru a testa citire corectă, nu doar `if ((scanf(...))` (pentru că şi `EOF` e nenul)

`fgets` returnează `NULL` dacă fişierul se termină înainte de a citi ceva
Exemplu: prelucrarea unui fişier linie cu linie

```
char lin[128]; while (fgets(lin, 128, stdin)) /* prelucrează lin */
```


Citirea anumitor categorii de caractere

Atenţie: Funcţiile din `ctype.h` returnează *nenul* pentru un caracter de felul dorit şi 0 în caz contrar (*NU* neapărat 1 şi 0)
nu scrieţi niciodată `if (isalpha(c) == 1)` ci doar `if (isalpha(c))`

Atenţie la cicluri infinite pentru sfârşit de fişier:

```
int c; while (isdigit(c = getchar())) /* ceva */
```

va ieşi din ciclu când `c` nu e cifră, inclusiv la `EOF` (nu e cifră)

```
int c; while (!isdigit(c = getchar())) /* ceva */
```

se va bloca la `EOF`, pentru că nu e cifră (nici `isalpha, isspace`, etc.)

```
while (!isdigit(c = getchar()))
  if (c == EOF) break; /* sau ce vrem să facem la EOF */
  else /* restul prelucrării */
```

Ignoră oricâte spaţii albe: `scanf(" ");`

Ignoră până la sfârşit de linie: `scanf("%*[^\n]"); scanf("%*1[\n]");`
(citeşte şi ignoră (`*`) oricâte caractere diferite (`^`) de `\n` şi un `\n`

Citirea şi prelucrarea până la sfârşit de fişier

Pentru a prelucra corect până la `EOF` nu e suficientă secvenţa din stânga deoarece `!feof()` la începutul ciclului nu garantează o citire corectă.
Trebuie testată citirea corectă (`getchar() != EOF`, `fgets(...) != NULL`, valorea lui `(f)scanf`), şi tratat cazul de eroare (ex. ieşirea din buclă)

```
while !(feof(fisier)) {
  citeste();
  prelucreaza();
}
```

```
for (;;) {
  if (citeste() != CORECT) break;
  prelucrează();
}
```

Indicatorul de sfârşit de fişier e poziţionat doar când se încearcă citirea *dincolo* de sfârşitul fişierului, nu când s-a citit ultimul caracter.
– după citirea ultimului element, `feof()` poate fi adevărat sau nu
ex. pt. un fişier de întregi separaţi prin spaţii, `feof()` e poziţionat după citirea ultimului doar dacă nu e urmat de altceva (ex. spaţiu, `\n`)
ex. la citirea linie cu linie, `feof()` e poziţionat după citirea ultimei linii doar dacă ea nu se termină cu `\n`.
– dacă `feof()` e fals, fişierul poate să mai conţină un element sau nu

Problema de la laborator

Să se tipărească, pe câte o linie, toate secvenţele de cifre din intrare.
O abordare: textul e o repetitie de: grup de cifre, grup de alte caractere
⇒ structura: două cicluri consecutive, într-un ciclu pana la EOF
`'\n'` se tipareste la trecerea intre cele doua (preferabil dupa cifre)
se incepe cu grupul de alte caractere (posibil vid)

```
void main(void)
{
  int c;
  do { /* si EOF e !isdigit, atentie la ciclu infinit! */
    while (!isdigit(c = getchar())) if (c == EOF) return;
    /* aici, c e sigur o cifra; repeta cat timp e cifra */
    do putchar(c); while (isdigit(c = getchar()));
    putchar('\n'); /* gata cifrele, c e altceva, poate EOF */
  } while (c != EOF);
}
```


Extragerea secventelor de cifre (cont.)

Mai simplu: când vedem o cifră, citim şi tipărim cât timp e cifră

```
void main(void) {
  int c;
  while ((c = getchar()) != EOF)
    if (isdigit(c)) { /* prima cifra; continua cu restul */
      do putchar(c); while (isdigit(c = getchar()));
      putchar('\n');  /* gata grupul de cifre */
    } /* daca nu e cifra, nu trebuie facut nimic */
}
```

De considerat, pentru prelucrări pe secvenţe de caractere:
– cum e definită secvenţa căutată, şi ce poate fi între secvenţe ?
– ce trebuie făcut la separarea între secvenţe (cicluri în program) ?
– care e starea (caracterul curent) înainte şi după fiecare ciclu ?
– `EOF` poate interveni oricând. Se opreşte corect programul ?
Încercaţi să priviţi problema (şi soluţia) ca un automat:
în orice punct din program, ce poate interveni ? ce trebuie făcut ?