

## FUNCȚII DE INTRARE – IEȘIRE PENTRU CARACTERE ȘI ȘIRURI DE CARACTERE

### 1. Funcții de intrare (citire)

#### 1.1. **int getc(FILE \*stream);**

- citește un caracter dintr-un fișier deschis pentru citire sau de la dispozitivul standard de intrare (stdin – de obicei asociat tastaturii) și incrementează indicatorul de poziție al fișierului pentru a adresa următorul caracter

- are prototipul în `STDIO.H`;

- returnează în caz de succes, caracterul citit (convertit în `int` fără extensie de semn) sau EOF pentru end-of-file sau eroare.

#### 1.2. **int getchar(void);**

- citește un caracter de la stdin (dispozitiv standard de intrare);

- are prototipul în `STDIO.H`;

- este o macrodefiniție (definit ca `getc(stdin)`) care returnează următorul caracter de la stdin;

- în caz de succes, `getchar()` returnează caracterul citit, după ce l-a convertit într-un `int` fără extensie de semn. Pentru end-of-file (sfârșit de fișier) sau eroare, returnează EOF (definit în `STDIO.H` ca -1). Caracterul CTRL-Z (0x1A) este interpretat ca EOF și convertit în -1 ;

- funcția lucrează bufferat (line buffered), adică nu putem prelucra caracterele citite până când nu s-a tastat Enter; caracterele introduse sunt memorate într-un buffer și abia după ce se tastează Enter, funcția le citește de acolo.

#### 1.3. **char \*fgets(char \*s, int n, FILE \*stream);**

- citește un șir de caractere dintr-un fișier deschis pentru citire sau de la tastatură în cazul în care **stream** este **stdin**;

- are prototipul în `STDIO.H`;

- funcția un șir de caractere terminat cu Enter de la tastatură sau o linie dintr-un fișier. Citirea se oprește la sfârșitul liniei sau la citirea a **n-1** caractere (se ia în considerare prima condiție îndeplinită). Funcția păstrează în șir caracterul `\n`, după care adaugă `\0`.

- șirul citit se stochează în **s**

- funcția returnează adresa șirului de caractere citit în caz de citire corectă, valoarea 0 (zero) la citirea sfârșitului de fișier și în caz de eroare.

- spre deosebire de funcția **gets** (a cărei descriere urmează), funcția **fgets** controlează și lungimea șirului de caractere, nepermițând depășirea lungimii specificate de programator. Din acest motiv se recomandă folosirea acestei funcții și nu a funcției **gets**.

## 2. Funcții de ieșire (scriere)

### 2.1. **int putc(int c, FILE \*stream);**

- scrie un caracter într-un flux (stdout sau un fișier deschis în scriere);
- are prototipul în STDIO.H;
- returnează caracterul c pentru succes, altfel EOF.

### 2.2. **int putchar(int c);**

- trimite un caracter la stdout (dispozitiv standard de ieșire);
- are prototipul în STDIO.H; este definit ca: `putc(c, stdout)`;
- returnează caracterul c pentru succes, EOF pentru eroare.

### 2.3. **int fputs(const char \*s, FILE \*stream)**

- scrie un șir de caractere terminat cu `\0` într-un fișier (sau pe monitor când fișierul indicat (stream) este stdout). Se presupune că șirul are deja în componență `\n`;
- are prototipul în STDIO.H;
- returnează ultimul caracter scris, iar în caz de eroare returnează EOF.

### 2.4. **int puts(const char \*s);**

- trimite un șir de caractere la stdout;
- are prototipul în STDIO.H;
- funcția copie șirul s (terminat cu NULL) la stdout, adăugând un newline (linie nouă);
- returnează o valoare pozitivă pentru succes, altfel EOF.

## 3. Funcții pentru clasificarea caracterelor

În continuare sunt prezentate câteva macrodefiniții din CTYPE.H, care ne ajută să diferențiem între ele caracterele.

### (a) **int isalpha(int c);**

- returnează un întreg diferit de zero, dacă c este literă (a ... z sau A ... Z).

**(b) int isascii(int c);**

– returnează un întreg diferit de zero dacă octetul inferior al lui c este în intervalul 0 ... 127 (0x00 ... 0x7F)

**(c) int iscntrl(int c);**

– returnează un întreg diferit de zero dacă c este caracterul Del (0x7F) sau un caracter de control obișnuit ce are codul ASCII în gama 0x00 ... 0x1F

**(d) int isdigit(int c);**

– returnează un întreg diferit de zero dacă c este o cifră zecimală (0 ... 9)

**(e) int islower(int c);**

– returnează un întreg diferit de zero dacă c este literă mică (a ... z).

**(f) int isspace(int c);**

– returnează un întreg diferit de zero dacă c este spațiu(' '), tab('\t'), carriage return('\r'), new line('\n'), vertical tab('\v') sau form feed('\f').

**(g) int isupper(int c);**

– returnează un întreg diferit de zero dacă c este literă mare (A ... Z).

**(h) int isxdigit(int c);**

– returnează un întreg diferit de zero dacă c este cifră hexazecimală (0 ... 9, a ... f, A ... F).

## TEMA 1

### Problema 1.1

Scrieți un program care număra aparițiile unui anumit caracter într-o secvență de caractere citită de la tastatură.

### Problema 1.2

Să se scrie un program care contorizează toate aparițiile vocalelor (fiecare în parte) dintr-o secvență citită de la tastatură. Se va include posibilitatea reluării programului la cerere.

**Problema 1.3**

Scrieți un program care folosește două funcții proprii care convertesc textul dintr-un fișier în litere mari sau litere mici după cum se indică prin dialog de la tastatură. Nu se va face apel la funcțiile de conversie din bibliotecă.

**Problema 1.4**

Să se scrie un program care afișează lungimile cuvintelor dintr-o secvență citită de la tastatură, secvența încheiata cu Enter. Se consideră că orice cuvânt se termină la primul caracter alb. Nu se vor folosi funcțiile de bibliotecă pentru lungimea unui șir.

**TEMA 2****Problema 2.1**

Să se scrie o funcție care are un parametru de tip unsigned long care reprezintă un număr dat în baza 10 și care calculează reprezentarea valorii parametrului funcției în baza 16 și depune rezultatul într-un șir de caractere dat ca al doilea parametru al funcției (șirul de caractere va începe cu 0x sau 0X).

Să se testeze funcția scrisă într-un program care citește un număr întreg de la tastatură și afișează șirul de caractere corespunzător reprezentării în baza 16.

(Nu se vor folosi eventualele funcții de conversie existente în bibliotecile mediului).

**Problema 2.2**

Se citește de la tastatură un text format din mai multe linii până la întâlnirea combinației de taste CTRL/Z (Windows) sau CTRL/D (Linux).

Stabiliți numărul de litere, cifre, numărul de separatori, numărul de cuvinte, numărul de propoziții (o propoziție se termină cu .) și numărul de aliniate (un aliniat începe cu un TAB).

De asemenea, se vor afișa numărul total de caractere și procente de litere și cifre din numărul total de caractere.

Cuvintele sunt separate prin caractere albe, punct, virgulă, două puncte, punct virgulă și cratimă. La sfârșitul fiecărei propoziții avem un punct urmat de un spațiu, iar un aliniat este terminat cu punct urmat de sfârșit de linie.

*Indicație:*

Textul se citește caracter cu caracter. Caractere albe sunt spațiu, TAB, Enter.

Nu se vor folosi alte funcții de bibliotecă decât cele pentru citire și scriere.