

Noțiuni de bază în programare

Obiectiv: fixarea noțiunilor de bază asociate cu codificarea și logica programelor.

Activități: prezentarea următoarelor subiecte:

- sistemele de numerație folosite de limbajul C - baza 2, 8, 16;
- avantajele utilizării bazei 16 în domeniul calculatoarelor;
- unități de măsură pentru cantități de date – Kilo/Mega/Giga/Terra bit/byte;
- tipuri de date fundamentale pentru calculatoare - bit, octet, word, double word;
- codificarea ASCII;
- scheme logice, simbolurile utilizate în schemele logice;

1.1 Sistemele de numerație folosite de limbajul C

Un număr poate fi reprezentat în mai multe sisteme de numerație (în mai multe baze). Reprezentarea cea mai folosită în viața de zi cu zi este cea **zecimală** (folosind baza 10), reprezentarea valorilor făcându-se folosind cele 10 cifre (de la 0 la 9).

Suportul matematic folosit de computere pentru manevrarea și prelucrarea informațiilor este numerația în baza 2, dar și în baze de numerație care sunt puteri ale lui 2.

Principalele sisteme de numerație pentru reprezentarea binară a informației sunt prezentate în continuare.

Sistemul binar folosește *baza 2*, fiind utilizate doar două cifre - 0 și 1. Astfel, orice număr va fi reprezentat numai cu aceste două cifre.

Sistemul octal folosește *baza 8*, utilizând cifrele de la 0 la 7. Astfel, numărul 8 va fi notat în sistemul octal cu 10, numărul 9 va fi notat cu 11, numărul 10 va fi notat în octal cu 12 etc. Sistemul octal este cel mai rar folosit.

Sistemul hexazecimal folosește *baza 16* - sunt necesare 16 cifre distincte, și după cifrele de la 0 la 9 se folosesc, în ordine, literele A, B, C, D, E, F. Astfel, numărul zecimal 15 va fi notat cu F, iar numărul 16 cu 10 (se citește “unu zero”), numărul zecimal 255 va avea notația hexa (prescurtare de la hexazecimal) FF ș.a.m.d.

Trecerea din baza 10 în baza 2 se face prin împărțiri succesive a numărului care se dorește transformat la 2, reținându-se de fiecare dată restul împărțirilor.

Exemplu de trecere din baza 10 în baza 2:

$$\begin{aligned} 29_{(10)} \Rightarrow & \quad 29 / 2 = 14 \quad (+ \underline{1}_5 \text{ rest}) \quad 14 / \\ & \quad 2 = 7 \quad (+ \underline{0}_4 \text{ rest}) \\ & \quad 7 / 2 = 3 \quad (+ \underline{1}_3 \text{ rest}) \quad 3 / 2 = \underline{1}_1 \quad (+ \underline{1}_2 \text{ rest}) \quad \Rightarrow 29 \text{ în baza } 2 \\ & \quad = 11101_{(2)} \end{aligned}$$

1 2 3 4 5

Trecerea din baza 2 în baza 10 se face prin înmulțiri succesive ale cifrelor numărului în baza 2 cu puteri ale lui 2:

Exemplu de conversie din baza 2 în baza 10:

$$11101_{(2)} = \underline{1} * 2^4 + \underline{1} * 2^3 + \underline{1} * 2^2 + \underline{0} * 2^1 + \underline{1} * 2^0 = 16 + 8 + 4 + 0 + 1 = 29_{(10)}$$

Introducere în Programarea calculatoarelor

Laborator 1

Din orice bază de numerație se poate trece în baza 10 prin înmulțiri succesive, iar din baza 10 se poate trece în orice bază de numerație prin împărțiri succesive.

În tabelul următor sunt prezentate cifrele de la 0 la 15 (în sistemul zecimal) și reprezentările lor în bazele prezentate:

Binar	Octal	Zecimal	Hexa
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

Tabelul 1.1 Reprezentarea primelor 15 numere în sistemele de numerație binar, octal, zecimal și hexazecimal

Exemplu de conversie din baza 2 în baza 16 (o cifra hexazecimală ocupa 4 biți)

$$10001010_{(2)} = 8A_{(16)} \quad (1000 = 8, 1010 = A)$$

Exemplu de conversie din baza 16 în baza 2:

$$B5_{(16)} = 10110101_{(2)} \quad (B = 1011, 5 = 0101)$$

Introducere în Programarea calculatoarelor

Laborator 1

1.2 Unități de măsură pentru cantități de date

Cantitatea de informație stocată și vehiculată de computer (în format binar) este măsurată în unități de măsură specifice. *O cifră binară* reprezintă un bit de informație și aceasta este unitatea de bază pentru măsurarea informației.

Din motive practice, informațiile sunt manevrate în grupuri de câte 8 biți. Un grup de 8 biți se numește *octet* sau *Byte*. (1 Byte reprezintă, pentru computer, cantitatea de memorie necesară pentru memorarea unei litere de text). Notățiile prescurtate fac diferența între bit (notat cu “b”) și Byte (notat cu “B”).

Aceste unități fiind mult prea mici pentru necesitățile actuale de stocare/prelucrare/etc., în majoritatea cazurilor practice se folosesc multiplii lor. Acești multipli sunt “Kilo”, “Mega”, “Giga” și “Terra”, iar valorile lor, relativ la valoarea unui byte, sunt:

$$\begin{aligned}
 1 \text{ KiloByte} &= 1.024 \text{ Bytes} &= 2^{10} \text{ bytes} \\
 1 \text{ MegaByte} &= 1.024 \text{ KiloBytes} = 1.024 * 1.024 \text{ Bytes} &= 2^{20} \text{ bytes} \\
 1 \text{ GigaByte} &= 1.024 \text{ MegaBytes} = 1.024 * 1.024 \text{ KiloBytes} &= 2^{30} \text{ bytes} \\
 1 \text{ TerraByte} &= 1.024 \text{ GigaBytes} = 1.024 * 1.024 \text{ MegaBytes} &= 2^{40} \text{ bytes}
 \end{aligned}$$

Multiplii cantităților de informații sunt prezentați în tabelul următor:

Prefix	Biți			Bytes (octeți)		
	Multiplu	Notăție	Valoarea	Multiplu	Notăție	Valoarea
Kilo	Kilobit	Kb	1024 biți	KiloByte	KB	1024 Bytes
Mega	Megabit	Mb	1024 Kb	MegaByte	MB	1024 KB
Giga	Gigabit	Gb	1024 Mb	GigaByte	GB	1024 MB
Terra	Terrabit	Tb	1024 Gb	TerraByte	TB	1024 GB

Tabelul 1.2 Multiplii unităților de măsură a datelor

În general, multiplii bit-ului sunt folosiți în măsurarea vitezelor de transmitere a informațiilor, iar multiplii byte-ului sunt folosiți în măsurarea capacităților de stocare/memorare.

Introducere în Programarea calculatoarelor

Laborator 1

1.3 Tipuri de date fundamentale pentru calculatoare

Calculatoarele pot lucra cu următoarele tipuri fundamentale de date: bit, byte, word, double word. Aceste tipuri de date sunt considerate tipuri fundamentale, deoarece un procesor poate lucra nativ (direct) cu informații reprezentate pe aceste tipuri de date.

Domeniul de valori pe care le poate lua o dată dintr-un tip este de $[0 \dots 2^N - 1]$, unde N este numărul de biți pe care se reprezintă tipul respectiv.

Caracteristicile tipurilor fundamentale sunt prezentate în tabelul următor:

Tipul datelor	Domeniul de valori	Lungimea (biți)	Lungimea (octeți)
Bit	0, 1	1	1/8
Byte	0 ... 255 (00h – FFh)	8	1
Word	0 ... 65.535 (0000h – FFFFh)	16	2
double word	0 ... 4.294.967.295 (00000000h – FFFFFFFFh)	32	4

Tabelul 2.3 Caracteristicile principalelor tipuri de date

1.4 Codificarea ASCII

ASCII este acronimul pentru *American Standard Code for Information Interchange* (Codul American Standard pentru Schimbul de Informații).

ASCII reprezintă o tabelă de asocieri între caractere și coduri numerice și a apărut deoarece calculatoarele lucrează doar cu numere, iar utilizatorii au nevoie și de litere și simboluri grafice. Fiecare literă și simbol grafic care se regăsește pe tastatură sau pe monitor are un cod corespunzător în tabela ASCII. În interiorul unui calculator, se lucrează doar cu aceste coduri asociate.

Setul standard de caractere ASCII conține 128 numere zecimale (0 - 127), care sunt asociate cifrelor, numerelor, semnelor de punctuație și celor mai utilizate caractere speciale.

Setul extins de caractere ASCII conține, de asemenea, 128 de numere zecimale (128 – 255) și conține caractere adiționale – simboluri matematice, grafice etc.

Cele mai importante caractere din setul *standard de caractere ASCII* sunt prezentate în continuare împreună cu reprezentările lor corespunzătoare în sistemele de numerație binar, zecimal, hexazecimal și octal:

Introducere în Programarea calculatoarelor

Laborator 1

Zecimal	Octal	Hex	Binar	Valoare	Descriere
000	000	000	00000000	NUL	(Null)
013	015	00D	00001101	CR	(Enter)
027	033	01B	00011011	ESC	(Escape)
032	040	020	00100000	SP	(Space)
033	041	021	00100001	!	
034	042	022	00100010	"	
035	043	023	00100011	#	
036	044	024	00100100	\$	
037	045	025	00100101	%	
038	046	026	00100110	&	(ampersant)
039	047	027	00100111	'	
040	050	028	00101000	(
041	051	029	00101001)	
042	052	02A	00101010	*	(asterisc)
043	053	02B	00101011	+	
044	054	02C	00101100	,	
045	055	02D	00101101	-	
046	056	02E	00101110	.	
047	057	02F	00101111	/	
048	060	030	00110000	0	
049	061	031	00110001	1	
050	062	032	00110010	2	
051	063	033	00110011	3	
052	064	034	00110100	4	
053	065	035	00110101	5	
054	066	036	00110110	6	
055	067	037	00110111	7	
056	070	038	00111000	8	
057	071	039	00111001	9	
058	072	03A	00111010	:	
059	073	03B	00111011	;	
060	074	03C	00111100	<	
061	075	03D	00111101	=	
062	076	03E	00111110	>	
063	077	03F	00111111	?	
064	100	040	01000000	@	
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	
079	117	04F	01001111	O	

Introducere în Programarea calculatoarelor

Laborator 1

080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[
092	134	05C	01011100	\	
093	135	05D	01011101]	
094	136	05E	01011110	^	
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	
107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	
113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	
124	174	07C	01111100		
125	175	07D	01111101	}	
126	176	07E	01111110	~	
127	177	07F	01111111	DEL	

Introducere în Programarea calculatoarelor

Laborator 1

În continuare se ilustrează *reprezentarea tabelară* a setului standard de caractere ASCII:

ZEC	HEX	CHR	ZEC	HEX	CHR	ZEC	HEX	CHR	ZEC	HEX	CHR
0	00	NUL	32	20	SP	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(72	48	H	104	68	h
9	09	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Tabelul 1.4 Reprezentarea setului standard de caractere ASCII

Introducere în Programarea calculatoarelor

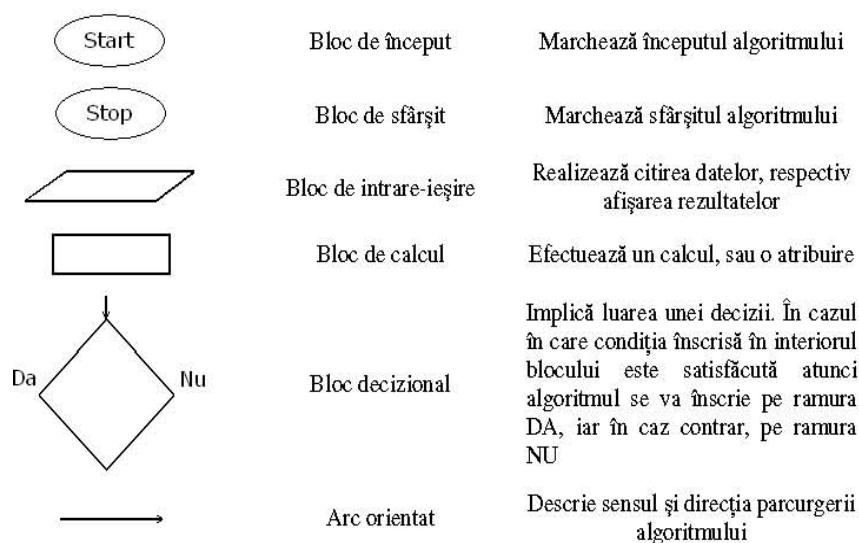
Laborator 1

ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character
128	Ç	160	à	192	Ł	224	α
129	ü	161	í	193	ł	225	β
130	é	162	ó	194	Ł	226	Γ
131	â	163	ú	195	ł	227	π
132	ä	164	ñ	196	—	228	Σ
133	ã	165	Ñ	197	+	229	σ
134	å	166	â	198	ƒ	230	μ
135	ç	167	ç	199	ƒ	231	τ
136	ê	168	ç	200	ƒ	232	Φ
137	ë	169	ç	201	ƒ	233	θ
138	è	170	ç	202	ƒ	234	Ω
139	í	171	½	203	ƒ	235	6
140	ï	172	¼	204	ƒ	236	×
141	ì	173	í	205	ƒ	237	Ø
142	Ä	174	«	206	ƒ	238	€
143	Å	175	»	207	ƒ	239	ƒ
144	Ê	176	⋯	208	ƒ	240	ƒ
145	æ	177	⋯	209	ƒ	241	ƒ
146	Æ	178	⋯	210	ƒ	242	ƒ
147	ô	179	—	211	ƒ	243	ƒ
148	ö	180	—	212	ƒ	244	ƒ
149	ó	181	ƒ	213	ƒ	245	ƒ
150	û	182	ƒ	214	ƒ	246	ƒ
151	ü	183	ƒ	215	ƒ	247	ƒ
152	ÿ	184	ƒ	216	ƒ	248	ƒ
153	Û	185	ƒ	217	ƒ	249	ƒ
154	Ü	186	ƒ	218	ƒ	250	ƒ
155	é	187	ƒ	219	ƒ	251	√
156	£	188	ƒ	220	ƒ	252	n
157	₣	189	ƒ	221	ƒ	253	2
158	₤	190	ƒ	222	ƒ	254	■
159	ƒ	191	ƒ	223	ƒ	255	(blank 'FF')

Tabelul 1.5 Setul extins de caractere ASCII

1.5 Scheme logice

Schemele logice sunt o “reprezentare” schematică a unui program. Ele sunt valabile în orice limbaj de programare, folosirea lor permițând o dezvoltare și o vizualizarea mai ușoară a programului. O *schemă logică* are în componență următoarele blocuri:



Introducere în Programarea calculatoarelor

Laborator 1

Exemplu 1 Schemă logică asociată cu calculul sumei a 2 numere:

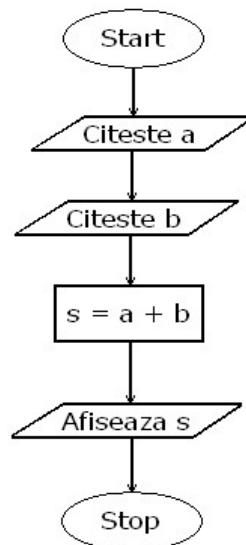


Figura 1.1 Schemă logică - suma a 2 numere

Exemplu 2 Schemă logică asociată cu calculul maximului a 2 numere:

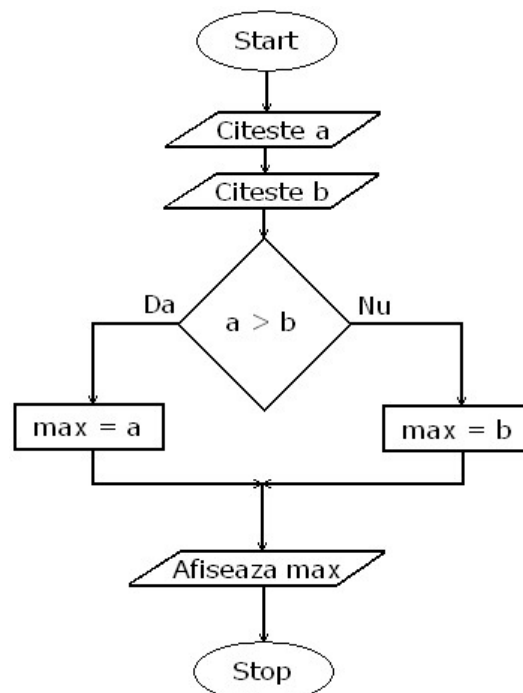


Figura 1.2 Schema logica - maximul a 2 numere

Exemplu 3 Schemă logică asociată cu rezolvarea unei ecuații de gradul I:

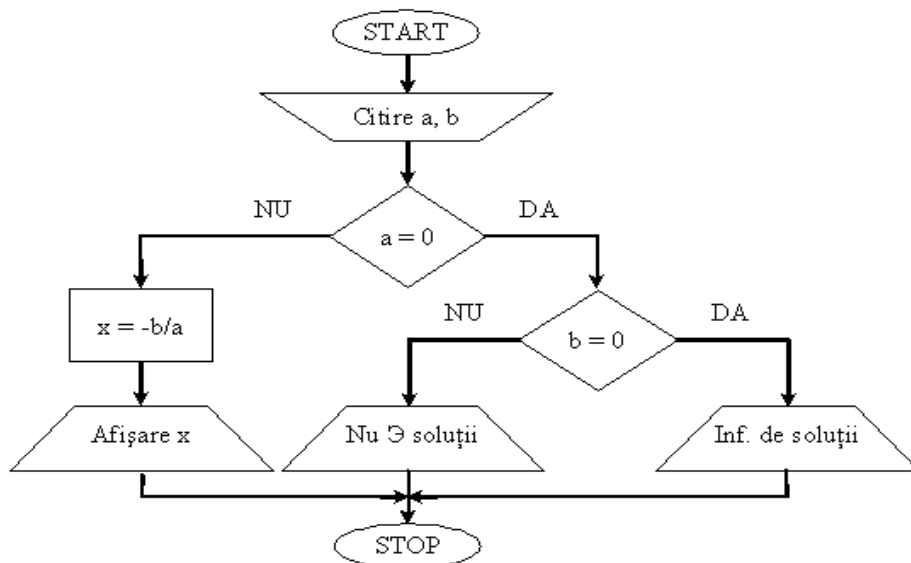


Figura 1.3 Schema logică – rezolvarea ecuației de grad I

Exemplu 4 Schemă logică asociată cu calculul sumei a n numere citite de la tastatură:

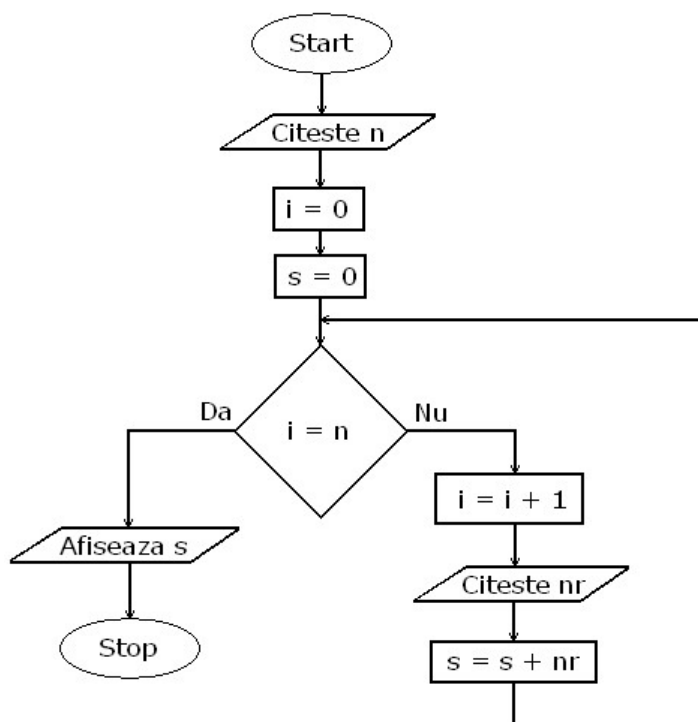


Figura 1.4 Schemă logică - suma a n numere citite de la tastatură

2.6 Probleme propuse

1. Realizați diverse conversii din/în bazele studiate (2 - 10, 10 - 2, 10 - 16, 16 - 10, 2 - 16, 16 - 2);
2. Realizați schema logică pentru determinarea maximului dintre trei numere: a, b și c.
3. Realizați schema logică pentru rezolvarea ecuației de gradul II.
4. Se citesc n numere. Realizați schema logică pentru determinarea sumei numerelor pare, respectiv a produsului numerelor prime.
5. Se dă un șir de numere care se citesc de la tastatură, citirea încheindu-se cu valoarea 0 (care nu face parte din șir). Să se afișeze perechile de numere consecutiv citite cu proprietatea că al doilea element al perechii este egal cu dublul primului.
6. Verificați dacă un număr este prim folosind o schemă logică.