Fundamentele programării
Curs 1

Șef lucr.dr.ing. GENGE Béla

Universitatea "Petru Maior", Departamentul de Informatică
Tîrgu Mureș, România
{bela.genge}@ing.upm.ro

Scop

- Introducere în gândirea computațională
- Dobândire cunoștințe algoritmică și limbaje de programare
- Înțelegerea codului sursă (altor coduri, reutilizare, plagiarism OK - cu limite)
- Înțelegerea scopului și limitelor limbajelor de programare (codarea problemelor științifice)

Administrativ

- Titular curs: **Șef lucr.(Lector) dr.ing. GENGE Béla**
- Titular laborator: **Asist.dr.ing. LEFKOVITS Szidónia**
- Pagina Web: `http://www.ibs.ro/~bela/`
- Adresa email: bela.genge@ing.upm.ro

- Examen: pe calculator (50% nota finală)
- Laborator: evaluări pe parcurs (50% nota finală)
- Condiție prezentare la examen: min. nota 5 lab. & max. 2 absențe lab.

De ce ar trebui să studiem programare?

- Asigurarea unui job la (înainte de) terminarea studiilor
- Profesia de dezvoltator software vine cu o serie de avantaje
- Trăim în era informațională - programarea dezvoltă gândirea

- Vizionare opinie profesioniști: `http://code.org`
- Exemplu de limbaj de programare (CodeHS): `http://code.org/learn/codehs`

Rolul calculatoarelor

- Calculatoarele ne ajută să mecanizăm, i.e., automatizăm, procesul de rezolvare a unei probleme
- Exemplu: algoritmul ce determină maximul dintre două numere a și b

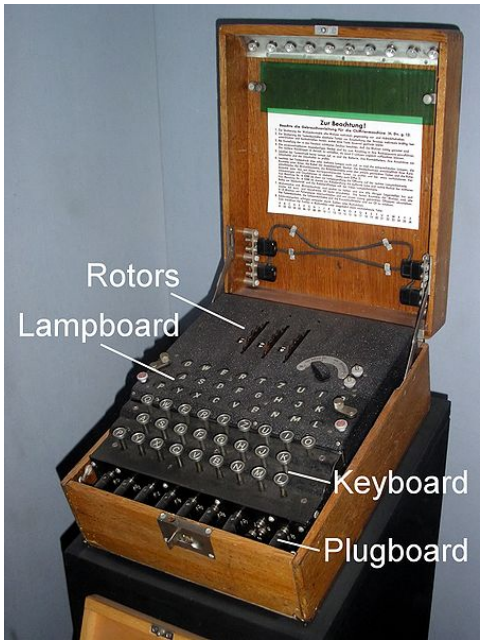
Algoritmul min-max

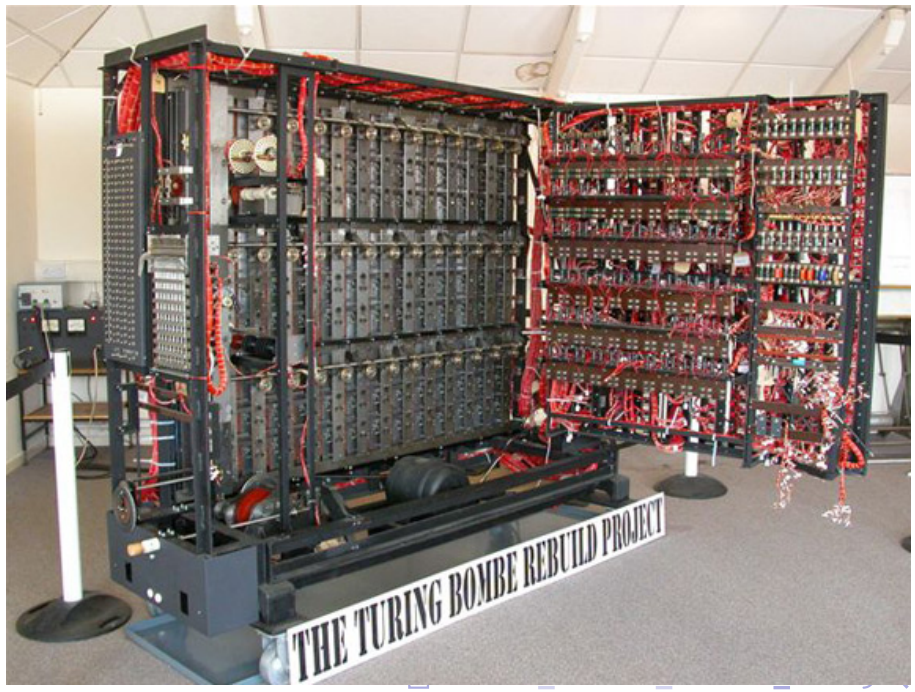
Intrare: a și b;
Ieșire: text;

```
Dacă a > b atunci
 text ⇐ "a > b";
Altfel
 text ⇐ "a <= b";
SfDacă
```

Implementarea calculatoarelor

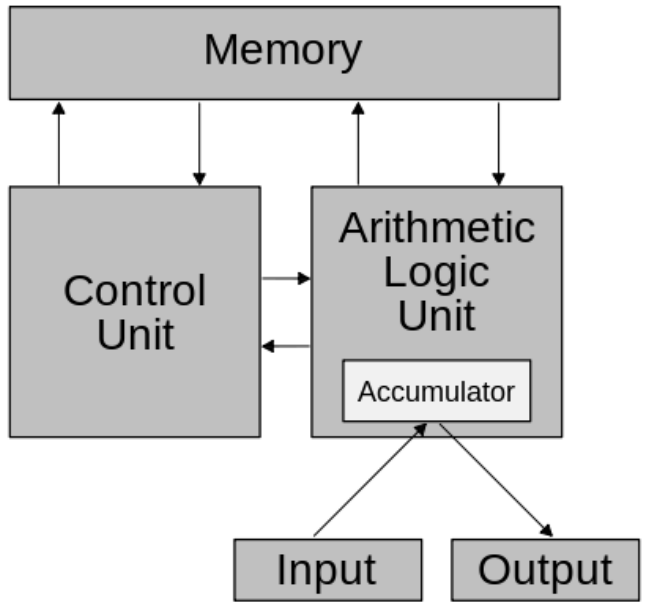
- Primele calculatoare: set de circuite ce rezolvă o anumită clasă de probleme
- Calculatoare dedicate: ceas, calculatorul de buzunar
- Exemplu: Mașina lui Turing (Bombe) construită pentru descifrarea codurilor criptate prin Enigma





Pasul următor: calculatoare generice

- Sunt instruite cum anume să rezolve o problemă
- Intrarea este extinsă cu un set de instrucțiuni
- Primele arhitecturi: Von Neumann (Neumann János Lajos) - 1945



De ce C/C++?

- Există mai mult de 100 limbaje de programare
- Adoptarea unui anumit limbaj depinde de scopul aplicației
- Un programator bun se descurcă cu orice limbaj (în limitele limbajului)
- De ce ar trebui să studiem C/C++:
  - Multe limbaje au fost influențate de C/C++ (Java, JavaScript, C#, Perl, Python, Ruby)
  - Majoritatea S.O. sunt implementate în C/C++
  - Codul mașină rezultat este foarte eficient

Despre ce vom discuta

- Elemente de bază (cuvinte rezervate, identificatori, constante, ...)
- Tipuri de date (scalare, derivate, tablouri, structuri, uniuni, ...)
- Instrucțiuni (if, switch, continue, goto, break)
- Funcții
- Operații pe biți
- Pointeri (alocare dinamică, pointeri la funcții)
- Șiruri de caractere
- Scheme logice

Resurse

- Brian W. Kernighan, Dennis M. Ritchie - The C programming language. 1978
- Adrian Runceanu - Programarea și utilizarea calculatoarelor. 2003
- Ion Cozac - Programare în limbajul C. 2004
- Vasile Petrovici - Programarea în limbajul C. 1993
- Delia Ungureanu, Adrian Ungureanu, Sorin Moraru - Programarea calculatoarelor; Îndrumar C++. 2001

Visual Studio și Primul Program

- Lansare Visual Studio. Cel mai simplu program
- Limbaj interpretat și compilat
- Drumul de la cod sursă la fișier executabil
- Afișare mesaj în C și în C++
- Comentarii

Tipuri de date

- Limbajul C permite stocarea valorilor în **constante** și **variabile**
- Tipul unei variabile determină:
  - Modul de reprezentare în memorie
  - Valorile pe care le poate lua
  - Timpul de viață
  - Dimensiunea zonei de memorie folosite
- Pentru declarațiile datelor sunt folosite două atribute:
  - Tipul unei date: standard sau definit de utilizator
  - Clasa de memorie: registru, stivă, heap

Clasificarea tipurilor de date

- Tipuri de bază

Type	Bits	Minimal Range
char	8	–127 to 127
unsigned char	8	0 to 255
signed char	8	–127 to 127
int	16 or 32	–32,767 to 32,767
unsigned int	16 or 32	0 to 65,535
signed int	16 or 32	Same as **int**
short int	16	–32,767 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	Same as **short int**
long int	32	–2,147,483,647 to 2,147,483,647
long long int	64	$-(2^{63}-1)$ to $2^{63}-1$ (Added by C99)
signed long int	32	Same as **long int**
unsigned long int	32	0 to 4,294,967,295
unsigned long long int	64	$2^{64}-1$ (Added by C99)
float	32	1E–37 to 1E+37 with six digits of precision
double	64	1E–37 to 1E+37 with ten digits of precision
long double	80	1E–37 to 1E+37 with ten digits of precision

- Tipuri derivate: **enumerate, structure, union, pointer**

Constante

- Constantele sunt valori nemodificabile pe parcursul execuției
- Dacă au asociate un identificator: constante simbolice
- Constante întregi: 12, 17, 20 (pot fi precizate și în alte baze de numerație)
- Constante caracter:
  - \0: Terminator de șir
  - \a: Generator de sunet
  - \b: Backspace
  - \n: Linie nouă
  - \r: Carriage return
  - \t: Tab orizontal
- Constante reale: 125.3, 32e3, -0.9
- Constante șir de caractere: "sirul meu", "folosire \" escape"
- Constante predefinite: CHAR_MAX, CHAR_MIN, INT_MAX, INT_MIN, ...

Segmente de memorie

- Zona de memorie a unui program C este organizată în 4 segmente principale:
  - Segmentul de cod: instrucțiunile programului
  - Segmentul de date: variabile globale și statice
  - Segmentul de stivă: variabile locale, date apel funcții
  - Segmentul de heap: zonă variabile alocate dinamic

Declarațiile de variabile

- **Variabilele** sunt datele care își pot modifica valoarea pe parcursul execuției.
- Sintaxa de declarare:

```
clasa_memorie tip lista_nume_variabile;
```

- **clasa_memorie**: auto, static, extern, register
- **tip**: tipuri de bază și utilizator
- **lista_nume_variabile**: listă cu denumiri alfa-numerice

Exemple de declarații

- Tipuri: int, char
- ASCII: American Standard Code for Information Exchange

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Alegerea corectă a variabilelor

- Valorile stocate
- Timpul de viață
- Eficiență program

- Exemple de declarări variabile pentru:
  - ora, minut, secunda și milisecunda
  - temperatura
  - rpm hard-disk

Problemă

- Cunoaștem previziunea temperaturii pentru ziua următoare sub forma valorii minime și maxime. Să se implementeze o aplicație ce determină valoarea temperaturii medie și o afișează ca un întreg.