Увод в програмирането

Лекция 8:

Работа с текст

(първа част)

Преговор

Мотивация

- Досега всички програми, които писахме, обработваха числа
- В реалния живот колко често използваме компютъра за математически цели? :)
- Следва да разгледаме обработката на текст
- Някои практични задачи, които ще можем да решим:
  - Търсене на дума в текст
  - Заместване на всички срещания на думата с друга дума
  - Анализ на текста – брой думи и т.н.

Представяне на текст в компютрите

- Компютърът работи с числа
- Символите (букви, цифри, знаци...) в един текст се представят с цели числа
- На всеки символ съответства конкретно число

```
It's easier
than you think.

49 74 27 73 20 65 61 73 69 65 72 0a 74 68
61 6e 20 79 6f 75 20 74 68 69 6e 6b 2e
```

(числата са показани в 16-ична бройна система за удобство)

ASCII таблица

- ASCII (American Standard Code for Information Interchange) таблица
- Указва на кой символ какво число съответства
- Не трябва да се учи наизуст :)

ASCII

- Кодове 0-127 (7 бита)
- За 128-255 има различни варианти Extended ASCII
  - Например за кирилица – Windows 1251
- Символите от 0 до 31 са по-специални
- Малките и главните английски букви имат различни кодове
- Кодовете на цифрите от 0 до 9 не са от 0 до 9
- Кодовете на съседни букви (цифри) също са съседни

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Допълнителен материал

- Ако имаме текст на немски, ще използваме една кодова таблица, ако имаме на гръцки – друга
- Какво става, ако искаме да напишем немско-гръцки разговорник?
- Очевидно 256 символа са крайно недостатъчни
- Най-доброто решение: Unicode
  - Един символ вече не е един байт – няма как
- Варианти: UTF-8 (най-често), UTF-16 и др.

Допълнителен материал

- Маймуни – защо виждаме ”маймуни” в някои уеб страници, неизползващи латиница?
  - Примери: Äîáðå äîøëè, P”PsP±CЂPµ
  - Защото не е зададена правилна кодова таблица
  - Когато се пише ръчно HTML, най-горе се указва с META таг какъв да бъде енкодингът
  - Ако не е указан, трябва от меню на браузъра да изберем правилния енкодинг
- SMS-ите на кирилица трябва да са по-кратки – защо?

Типът char

- Стойностите са символи
- 8-битов
- Може да се използва и като числов тип – unsigned char (0..255) и signed char(-128..127)
- Литерали – символът се огражда с единични апострофи
  - Примери: 'a', '1', ' ', '\n'

Основни операции

- Дефиниране на променлива от тип char:
  `char c = 'M';`
- Отпечатване:
  `cout << c; // отпечатва M без апострофите`
- Въвеждане от клавиатурата:
  `cin >> c; //чете символ, пропуска интервали, табулация и нов ред – също като четенето на числа`

Основни операции (2)

- Сравнение – по познатия ни начин

```
cout << "Do you want to continue? Y/N";
cin >> c;
if (c == 'y' || c == 'Y')
    cout << "Let's go!" << endl;
```

- Можем да използваме и <, >, <= и >= (например при сортиране по азбучен ред)

```
if (c >= 'a' && c <= 'z')
    cout << "Small letter";
```

- Аритметични операции – все пак е число

```
c = 'A'; c += 3; cout << c; // D
```

- Демонстрация – вж. файла с програмен код

Примери

- Преобразуване на малка буква в главна и обратно
  - По много готин начин, без да знаем наизуст ASCII таблицата

- Материалът, който се преподава на информатичните и математическите специалности, е леко различен
- Затова има две версии на втората част на презентацията –
  - Едната е предназначена само за Инф. и ИС,
  - Другата – само за математиците
- Естествено, за обща култура всеки може да види и слайдовете за другата специалност