

# 1 Тэмдэгт мөр

## 1.1 ASCII код

ASCII (American Standard Code for Information Interchange) нь Англи хэл дээр суурилсан, тодорхойлогдсон 128 тэмдэгтүүдийг (0-9, A-Z, a-z, цэг, таслал, асуух, анхааруулах тэмдэгтүүд, зай, хуучин холбооны машинуудад хэрэглэгддэг удирдлагын тэмдэгтүүд) 7-bit санах ойд хадгалах арга. ASCII код нь компьютерт текстийг дүрсэлнэ.

Орчин үеийн тэмдэгт дүрслэх аргууд нь ASCII код дээр суурилсан боловч өөр олон тэмдэгтүүдийг хадгалах чадвартай.

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	□

Зураг 1: ASCII хүснэгт

Компьютерын процессор, санах ойн хэмжээ томорсноор ASCII кодыг 8-bit (1 byte) санах ойд хадгалах болсон. Жижиг 'a' үсгийн ASCII код нь 97 байхад том 'A' үсгийн ASCII код нь 65 байна. Энэ нь эдгээр тэмдэгтүүд ялгаатай тэмдэгтүүд болохыг илтгэнэ.

## 1.2 Тэмдэгт

Нэг нэгж бүхий тэмдэгт, эсвэл үсэг бүхий мэдээллийг **тэмдэгт** гэнэ. Тэмдэгтийг Си хэлэнд дүрслэхдээ дан апостроф ( ' ' ) дотор бичнэ.

```
1 #include <stdio.h>
2 int main() {
3     printf("%d\n", 'A'); // A-ийн ASCII code
4     printf("%c\n", 'A'); // A тэмдэгт
5     printf("%c\n", 'aB'); // Aldaa: davhar тэмдэгт байж болохгүй
6     return 0;
7 }
```

Тэмдэгтийг Си хэлэнд *char* төрлийг ашиглан хадгална. *char* нь 1 byte бөгөөд 0-255 хүртэлх утгуудыг хадгалах чадвартай ба ASCII кодыг бүрэн *char* төрөлд хадгалж болно.

```
1 #include <stdio.h>
2 int main() {
3     char a = 65;
4     printf("%d\n", a); // ASCII code
5     printf("%c\n", a); // Тэмдэгт
6     return 0;
7 }
```

*char* төрлийг аравтын тооллын системээр хэвлэвэл ASCII кодыг нь хэвлэнэ. Харин тэмдэгтийг нь хэвлэвэл тухайн ASCII кодоод харгалзах хэвлэж болдог тэмдэгтийг хэвлэнэ.

### 1.2.1 Тусгай тэмдэгтүүд

Си хэлэнд зарим тэмдэгтүүд тусгай тэмдэглэгээнүүдээс тогтдог. Тэдгээр тэмдэгтүүд нь доорх утгыг илэрхийлнэ.

<code>\b</code>	backspace
<code>\f</code>	дэлгэц цэвэрлэх
<code>\n</code>	шинэ мөр (enter дарахтай адил)
<code>\r</code>	бичих курсорыг мөрийн эхэнд аваачих
<code>\t</code>	хөндлөн tab-ийн зай авах
<code>\v</code>	босоо tab-ийн зай авах
<code>\x</code>	<code>\x</code> тэмдэгтээс хойших тэмдэгтүүд 16-тын тооллын системд хэвлэгдэнэ
<code>"</code>	" тэмдэгтийг оруулах
<code>'</code>	' тэмдэгтийг оруулах

```
\\
    хойшоо зураасыг оруулах \
\\ddd
    энд ddd нь 8-тын тооллоор бичсэн ascii код
```

Жишээ нь:

```
1 #include <stdio.h>
2 int main()
3 {
4     char signal = '\\7';
5     printf("Beep!\\n");
6     printf("%c", signal);
7     printf("ch = '\\a\\' \\n");
8     printf(" <- Start of this line!! \\r");
9
10    return 0;
11 }
```

### 1.3 Тэмдэгт мөр

Үг, өгүүлбэр нь тэмдэгтүүдийг дараалал гэж хэлж болно. Тэмдэгт мөр Си хэлэнд давхар апостроф (" ") дотор бичигддэг.

```
1 #include <stdio.h>
2 int main() {
3     printf("%s", "C hel");
4     return 0;
5 }
```

Тэмдэгт мөрийн хамгийн сүүлийн элемент нь тэмдэгт мөр төгсгөл заагч (0 буюу '\\0') байдаг.

%s гэдэг нь тухайн хаягт байгаа тэмдэгтүүдийг тэмдэгт мөрийн төгсгөл заагч хүртэл нь хэвлэдэг.

Тэмдэгт мөрийг хадгалахдаа тэмдэгтэн хүснэгтийг ашиглан хадгалдаг.

```
1 #include <stdio.h>
2 int main() {
3     char s[10] = "C hel";
4     printf("%s\\n", s);
5     return 0;
6 }
```

s-ийн индекс	0	1	2	3	4	5
Тэмдэгтүүд	'C'	' '	'h'	'e'	'l'	'\\0'
ASCII код	67	32	104	101	108	0

#### 1.3.1 Тэмдэгт мөрийг гараас унших

Тэмдэгтийн цувааг гараас оруулахдаа дараах хоёр стандарт функцийг хэрэглэж болно.

1. scanf() функц

```
1 char s[SIZE];
2 scanf("%s", s);
```

Тэмдэгтийн цувааг хүснэгтийн эхнээс байрлуулан хамгийн эхэнд гарч ирсэн хоосон зай эсвэл шинэ мөрөнд шилжүүлэх тэмдэгт хүртэл оруулах ба оруулж дуусаад мөрийн төгсгөл заагчийг автоматаар хадгалж өгнө.

## 2. gets() функц

```
1 char s[SIZE];
2 gets(s);
```

Тэмдэгтийн цувааг хүснэгтийн эхнээс байрлуулах бөгөөд хамгийн эхэнд гарч ирсэн шинэ мөрөнд шилжүүлэх тэмдэгт хүртэл оруулах ба оруулж дуусаад мөрийн төгсгөл заагчийг автоматаар хадгалж өгнө.

### 1.3.2 Тэмдэгтийн мөрийг дэлгэцлэх

Тэмдэгтийн цувааг дэлгэцэд хэвлэх Си хэлний дараах хоёр стандарт функцийг хэрэглэж болно.

#### 1. printf() функц

```
1 printf("%s", s);
```

Тэмдэгтийн цувааг эхнээс нь төгсгөл заагч хүртэл нь дэлгэцэнд хэвлэнэ.

#### 2. puts() функц

```
1 puts(s);
```

Тэмдэгтийн цувааг эхнээс нь төгсгөл заагч хүртэл нь дэлгэцэнд хэвлээд заагчийг шинэ мөрөнд шилжүүлнэ.

### 1.3.3 Тэмдэгт мөртэй ажиллах функцүүд

Си хэлэнд тэмдэгтийн цуваатай ажиллах стандарт функцүүд <string.h> толгой файлд агуулагддаг.

```
1 size_t strlen(const char *str);
```

Цувааны уртыг олно.

```
2.
1 char *strcpy(char *destination, const char *source);
```

dst хаягт байгаа тэмдэгт мөрийг, src хаяг руу хуулах

```
3.
1 int strcmp(const char *str1, const char *str2);
```

```
4.
1 char *strcat(char *destination, const char *source);
```

*source* хаяг дээр байгаа тэмдэгт мөрийг, *destination* хаяг дээр байгаа тэмдэгт мөрийн төгсгөл заагчаас нь эхлэн залгуулан хуулна. *destination*-ны зааж буй хаягийг буцаана.

## 1.4 Тэмдэгт мөртэй ажиллах

Тэмдэгт мөр нь хүснэгтэд хадгалагдах учраас жирийн давталт, хүснэгтүүдийг ашиглан тэмдэг мөртэй харьцдаг.

Жишээ: Өгөгдсөн тэмдэгт мөрд хэдэн ширхэг том үсэг байгааг олох.

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char s[100];
5     gets(s);
6     int i;
7     int count = 0;
8     for (i = 0; i < strlen(s); i++)
9         if (s[i] >= 'A' && s[i] <= 'Z')
10             count++;
11
12     puts(s);
13     printf("%d\n", count);
14     return 0;
15 }
```

## 1.5 Дасгалууд

### 1.5.1 Ангид

1. Доорх асуултуудыг асуугаад, тухайн хүний намтрыг гаргадаг програм бич.

- (a) Таны овог?
- (b) Таны нэр?
- (c) Та төрсөн он сар өдөр?
- (d) Дуртай өнгө?
- (e) Хобби юу вэ?
- (f) Чөлөөт цагаараа юу хийдэг вэ?

.....овогтой .....нь .....өдөр төрсөн. Одоо .....настай. Түүний дуртай өнгө бол ..... Тэр чөлөөт цагаараа .....хийдэг. Түүний хобби бол .....

2. Тэмдэгтийн цуваа өгөгдөхөд тухайн тэмдэгтийн цуваанд хэдэн эгшиг байгааг ол.

3. Хоёр үг өгөгдөхөд яг адилхан үг эсэхийг шалга.

4. Дараах код s-ийг s2-д хуулан "hello" гэсэн хариу гаргах ёстой бол алдааг олж зас.

```
1 #include <stdio.h>
2 int main() {
3     char s[] = "hello";
4     char s2[20] = 'MUIS-iinhaan';
```

```
5   for (i = 0; i < 5; i++)
6       s2[i] = s[i];
7
8   printf("%s", s2);
9   return 0;
10 }
```

5. Өгөгдсөн 2 тэмдэгт мөрийг нийлүүлэн нэг тэмдэгтэн хүснэгтэд хэвлэж харуул.
6. Хоосон зай агуулсан өгүүлбэрийг хадгалж. Үг үгээр нь таслан хэвлэж харуул.

### 1.5.2 Гэрт

1. Тэмдэгтийн цуваа өгөгдөхөд тухайн тэмдэгтийн цуваанд хэдэн үг байгааг ол.
2. Өгөгдсөн өгүүлбэрээс хамгийн олон эгшигтэй үгийг ол.
3. Тэмдэгтийн цуваа өгөгдөөд дараагаар нь нэг үг өгөгдөхөд, тухайн үг тэмдэгтэн цувааны хэд дүгээр байрлал дээр олдож байг ол. Олдохгүй бол -1 гэж гарга.