




## **Лекція 9. Таблиця ASCII та її застосування**

---



Тип даних `char` - це цілочисельний тип даних, який використовується для представлення символів. Тобто, кожному символу відповідає певне число з діапазону  $[-128; 127]$ .

---

```
#include <stdlib.h>
#include <stdlib.h>
```

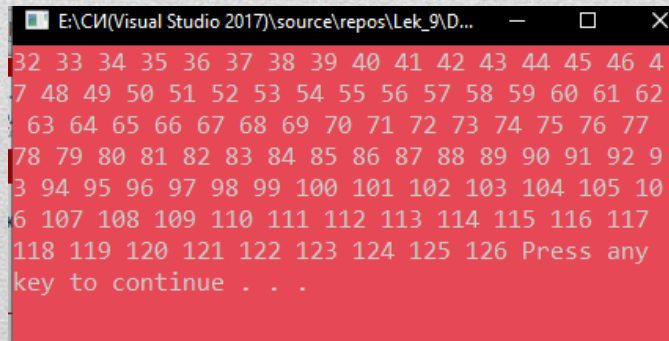
```
int main(){

char ch = 32;
while (ch < 127) {
    printf("%d ", ch);
    ch++;
}
system("pause");
return 0;
}
```

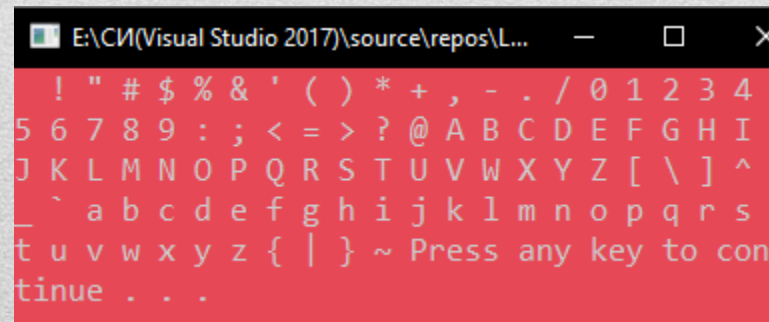
```
#include <stdlib.h>
#include <stdlib.h>
```

```
int main(){

char ch = 32;
while (ch < 127) {
    printf("%c ", ch);
    ch++;
}
system("pause");
return 0;
}
```



```
E:\CI(Visual Studio 2017)\source\repos\Lek_9\D...
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 4
7 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62
63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 9
3 94 95 96 97 98 99 100 101 102 103 104 105 10
6 107 108 109 110 111 112 113 114 115 116 117
118 119 120 121 122 123 124 125 126 Press any
key to continue . . .
```



```
E:\CI(Visual Studio 2017)\source\repos\L...
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4
5 6 7 8 9 : ; < = > ? @ A B C D E F G H I
J K L M N O P Q R S T U V W X Y Z [ \ ] ^
_ ` a b c d e f g h i j k l m n o p q r s
t u v w x y z { | } ~ Press any key to con
tinue . . .
```

ASCII (від англ. American Standard Code for Information Interchange) - американський стандартний код для обміну інформацією.

1	␣	25	↓	49	1	73	I	97	a	121	y	145	æ	169	┌	193	┐	217	┘
2	␣	26		50	2	74	J	98	b	122	z	146	Æ	170	└	194	┌	218	┘
3	♥	27		51	3	75	K	99	c	123	{	147	ø	171	┐	195	└	219	┘
4	♦	28	┌	52	4	76	L	100	d	124		148	ö	172	└	196	┐	220	┘
5	♣	29	↔	53	5	77	M	101	e	125	}	149	ò	173	┐	197	└	221	┘
6	♠	30	▲	54	6	78	N	102	f	126	~	150	û	174	«	198	┐	222	┘
7		31	▼	55	7	79	O	103	g	127		151	ù	175	»	199	└	223	┘
8		32		56	8	80	P	104	h	128		152	ÿ	176		200	┐	224	α
9		33	!	57	9	81	Q	105	i	129		153	Û	177		201	└	225	β
10		34	"	58	:	82	R	106	j	130		154	Ü	178		202	┐	226	γ
11		35	#	59	;	83	S	107	k	131		155	ç	179		203	└	227	δ
12		36	\$	60	<	84	T	108	l	132		156	£	180	┌	204	┐	228	ε
13		37	%	61	=	85	U	109	m	133		157	¥	181	└	205	┐	229	ζ
14		38	&	62	>	86	V	110	n	134		158	₣	182	┐	206	└	230	η
15		39	'	63	?	87	W	111	o	135		159	ƒ	183	└	207	┐	231	θ
16	▸	40	(	64	@	88	X	112	p	136	¡	160	á	184	┐	208	└	232	ι
17		41	)	65	A	89	Y	113	q	137	¢	161	í	185	└	209	┐	233	ϑ
18	‡	42	*	66	B	90	Z	114	r	138	£	162	ó	186	┐	210	└	234	ϒ
19	‡‡	43	+	67	C	91	[	115	s	139	¤	163	ù	187	┐	211	└	235	δ
20	‡‡‡	44	,	68	D	92	\	116	t	140	¥	164	ñ	188	┐	212	└	236	ε
21	‡‡‡‡	45	-	69	E	93	]	117	u	141	¦	165	Ñ	189	┐	213	└	237	ϕ
22	‡‡‡‡‡	46	.	70	F	94	^	118	v	142	§	166	ª	190	┐	214	└	238	ϕ
23	‡‡‡‡‡‡	47	/	71	G	95	_	119	w	143	¨	167	º	191	┐	215	└	239	ϕ
24	‡‡‡‡‡‡‡	48	0	72	H	96	`	120	x	144	©	168	¸	192	┐	216	└	240	ϕ

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

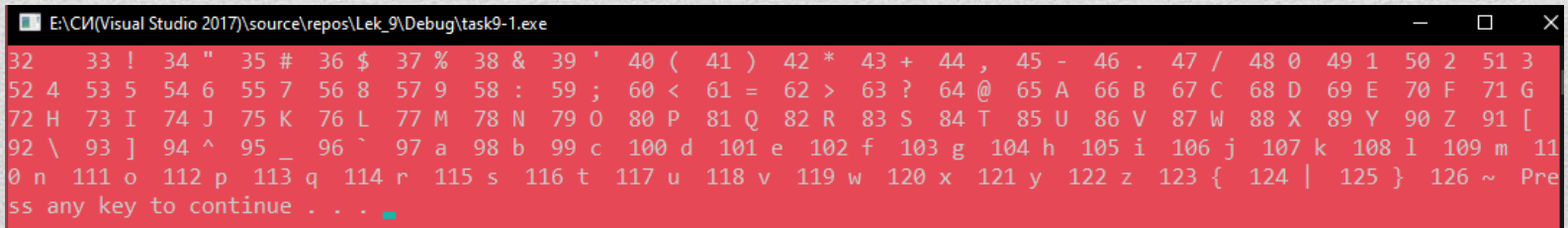
```
int main(){
```

```
    for (int code = 32; code < 127; code++)  
        printf("%d - %c\t", code, code );
```

```
    system("pause");
```

```
    return 0;
```

```
}
```



```
E:\C\C\Visual Studio 2017\source\repos\Lek_9\Debug\task9-1.exe  
32  33 !  34 "  35 #  36 $  37 %  38 &  39 '  40 (  41 )  42 *  43 +  44 ,  45 -  46 .  47 /  48 0  49 1  50 2  51 3  
52 4  53 5  54 6  55 7  56 8  57 9  58 :  59 ;  60 <  61 =  62 >  63 ?  64 @  65 A  66 B  67 C  68 D  69 E  70 F  71 G  
72 H  73 I  74 J  75 K  76 L  77 M  78 N  79 O  80 P  81 Q  82 R  83 S  84 T  85 U  86 V  87 W  88 X  89 Y  90 Z  91 [  
92 \  93 ]  94 ^  95 _  96 `  97 a  98 b  99 c  100 d  101 e  102 f  103 g  104 h  105 i  106 j  107 k  108 l  109 m  11  
0 n  111 o  112 p  113 q  114 r  115 s  116 t  117 u  118 v  119 w  120 x  121 y  122 z  123 {  124 |  125 }  126 ~  Pre  
ss any key to continue . . .
```



Дані типу `char` є цілими типом і займають в пам'яті комп'ютера 1 байт.

Співвідношення “символ-код” розміщується у таблиці символів Windows. Символи з кодами від 0 до 127 – це зарезервовані символи BIOS. Вони включають найбільш вживані символи, символи цифр, символи латинської абетки. **Ці СИМВОЛИ ЗМІНИТИ НЕМОЖНА.**

Символи з кодами від 128 до 255 – це регіональні символи, що прив'язані до конкретної абетки того комп'ютера на якому встановлена операційна система Windows.

---

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    char symbol;
    for (int i = 0; i < 180; i++) {
        symbol = (char)i;
        printf("%2d = %2c  ", i, symbol);
    }

    system("pause");
    return 0;
}
```

---



## Приклад. Що буде виведено на консоль?

```
#include <stdlib.h>
#include <stdio.h>
int main() {
    char ch = 3;
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++) {
            if (j > i) break;
            printf("%c ", ch);
        }
        printf("\n");
    }
    system("pause");
    return 0;
}
```

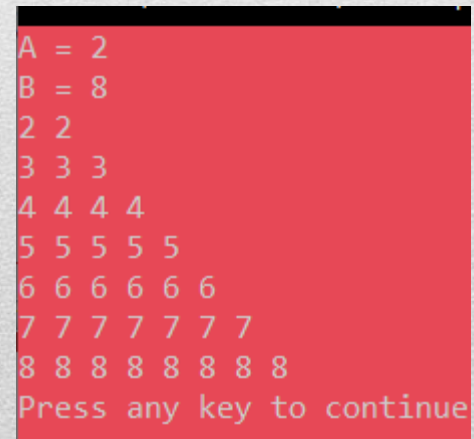


```
if(ch>='a' && ch <='z')  
if (ch >= 97 && ch <= 122)
```

---

**Приклад.** Дано цілі додатні числа  $A$  і  $B$  ( $A < B$ ). Вивести всі цілі числа від  $A$  до  $B$  включно; при цьому кожне число має виводитися стільки разів, яке його значення (наприклад, число 3 виводиться 3 рази).

```
int main() {
int A, B;
printf("A = ");    scanf("%d", &A);
printf("B = ");    scanf("%d", &B);
    for (int i = A; i <= B; i++) {
        for (int j = 1; j <= i; j++)
            printf("%d ",i);
        printf("\n");
    }
system("pause");
return 0;
}
```



```
A = 2
B = 8
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
Press any key to continue
```

**Приклад.** Дано цілі додатні числа  $A$  і  $B$  ( $A < B$ ).  
Вивести всі цілі числа від  $A$  до  $B$  включно; при цьому кожне число має виводитися стільки разів, яке його значення (наприклад, число 3 виводиться 3 рази).

```
int A, B;
do {
    system("cls");           // очищаємо вікно консолі
    printf("A = "); scanf("%d", &A);
    printf("B = "); scanf("%d", &B);
} while (A >= B);
for (int i = A; i <= B; i++) {
    for (int j = 1; j <= i; j++)
        printf("%d ", i);
    printf("\n");
}
```

## Приклад.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdlib.h>
#include <stdio.h>
int main() {
system("chcp 1251"); // перехід в консолі на укр. мову
int A, B, y=1;
do {
    do {
        system("cls"); // очищаємо вікно консолі
        printf("A = "); scanf("%d", &A);
        printf("B = "); scanf("%d", &B);
    } while (A >= B);
        for (int i = A; i <= B; i++) {
            for (int j = 1; j <= i; j++)
                printf("%d ", i);
            printf("\n");
        }
    printf("Продовжити-1, закінчити -0 "); scanf("%d", &y);
}while (y);
system("pause"); return 0;
}
```

**Приклад.** Послідовно вводяться числа. Після введення кожного числа потрібно вивести на екран, скільки серед них вже введено негативних. Закінчити введення, коли негативних стане > 5

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdlib.h>
#include <stdio.h>
int main() {
system("chcp 1251"); system("cls");
int y=1,x, count = 0;
do {
    do {
        printf("\n введіть число:"); scanf("%i", &x);
        if (x < 0) count++;
        printf("від'ємних чисел - %i", count);
    } while (count < 5);
    printf("\n Почати знову-1, закінчити -0 "); scanf("%d", &y);
}while (y);
system("pause");
return 0;
}
```

---

Приклад.

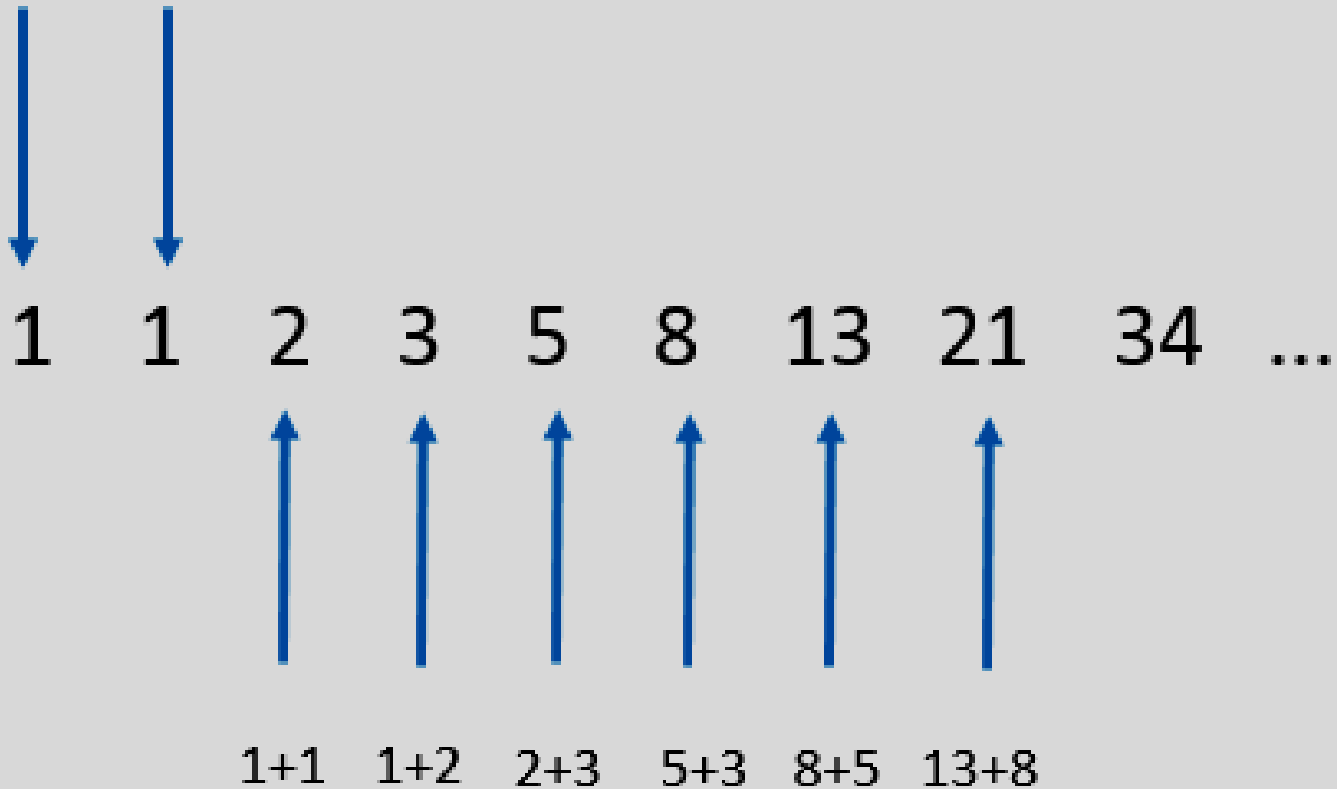
За допомогою циклів for надрукувати:

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

```
for (int i = 1; i <= 10; i++) printf("%3d", i);  
for (int i = 9; i > 0; i--) printf("%3d", i);
```

---

# Числа Фібоначчі





## Приклад. Вивести N чисел Фібоначчі.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdlib.h>
#include <stdio.h>
int main() {
system("chcp 1251"); system("cls");
int n,y=1;
do {
printf("Input n: "); scanf("%d", &n);
int fib = 0, fib1=0, fib2=1;
for (int i = 1; i <= n; i++){
fib = fib2 + fib1;
fib2 = fib1;
fib1 = fib;
printf("%d  ", fib);
}
printf("\n Почати знову-1, закінчити -0 "); scanf("%d", &y);
}while (y);
system("pause");
return 0;
}
```

---

## Приклад.

Дано ціле число  $N (> 1)$ , що є числом Фібоначчі. Знайти ціле число яке знаходиться перед і після нього.

```
int n, y=1;
do {printf(" 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,
377, 610, 987, 1597, 2584, 4181, 6765, 10946\n");

    printf("N:");scanf("%i", &n);
    int f1 = 1, f2 = 1, f = 0;
    while (f < n) {
        f = f2 + f1;
        f2 = f1;
        f1 = f;
    }
    printf("F_k-1=%i\nF_k+1=%i\n", f2, f1 + f2);

printf("\n Почати знову-1, закінчити -0 "); scanf("%d", &y);
}while (y);
```

---