



У мові C для символів визначено тип **char**.

Кожному символу відповідає число від 0 до 255, яке називається **ASCII-кодом** символу.

Наприклад, символу **'A'** відповідає число **65**.

Символами можна оперувати, як числами і, навпаки, змінні типу **int** можна інтерпретувати як символи (порівнювати з символами або друкувати як символи).



0 -	16 - ▶	32 -	48 - 0	64 - @	80 - P	96 - '	112 - p
1 - ☺	17 - ◀	33 - !	49 - 1	65 - A	81 - Q	97 - a	113 - q
2 - ☹	18 - ⇕	34 - "	50 - 2	66 - B	82 - R	98 - b	114 - r
3 - ♥	19 - !!	35 - #	51 - 3	67 - C	83 - S	99 - c	115 - s
4 - ♦	20 - ♠	36 - \$	52 - 4	68 - D	84 - T	100 - d	116 - t
5 - ♣	21 - ⚙	37 - %	53 - 5	69 - E	85 - U	101 - e	117 - u
6 - ♠	22 - ▬	38 - &	54 - 6	70 - F	86 - V	102 - f	118 - v
7 -	23 - ⇕	39 - '	55 - 7	71 - G	87 - W	103 - g	119 - w
8 -	24 - ↑	40 - (56 - 8	72 - H	88 - X	104 - h	120 - x
9 -	25 - ↓	41 -)	57 - 9	73 - I	89 - Y	105 - i	121 - y
10 -	26 - →	42 - *	58 - :	74 - J	90 - Z	106 - j	122 - z
11 -	27 - ←	43 - +	59 - ;	75 - K	91 - [107 - k	123 - {
12 -	28 - ⊥	44 - ,	60 - <	76 - L	92 - \	108 - l	124 -
13 -	29 - ⇄	45 - -	61 - =	77 - M	93 - j	109 - m	125 - }
14 - 🎵	30 - ▲	46 - .	62 - >	78 - N	94 - ^	110 - n	126 - ~
15 - ⚙	31 - ▼	47 - /	63 - ?	79 - O	95 - ÷	111 - o	127 - ☺
16 - ▶	32 -	48 - 0	64 - @	80 - P	96 -	112 - p	

128 - A	144 - P	160 - a	176 - ☼	192 - L	208 - ⊥	224 - p	240 - Ę
129 - Б	145 - C	161 - б	177 - ☼	193 - ⊥	209 - ⊥	225 - c	241 - ë
130 - B	146 - T	162 - в	178 - █	194 - ⊥	210 - ⊥	226 - т	242 - Ę
131 - Г	147 - Y	163 - г	179 -	195 - ⊥	211 - ⊥	227 - y	243 - e
132 - Д	148 - Ф	164 - д	180 -]	196 - —	212 - ⊥	228 - ф	244 - Ę
133 - E	149 - X	165 - e	181 -]	197 - +	213 - ⊥	229 - x	245 - i
134 - Ж	150 - Ц	166 - ж	182 -	198 - ⊥	214 - ⊥	230 - ц	246 - Ÿ
135 - З	151 - Ч	167 - з	183 -	199 - ⊥	215 - ⊥	231 - ч	247 - Ÿ
136 - И	152 - Ш	168 - и	184 - ¶	200 - ⊥	216 - ⊥	232 - ш	248 - °
137 - Й	153 - Щ	169 - й	185 -	201 - ⊥	217 - ⊥	233 - щ	249 - ●
138 - K	154 - Ъ	170 - к	186 - ⊥	202 - ⊥	218 - ⊥	234 - ъ	250 - .
139 - Л	155 - Ы	171 - л	187 - ⊥	203 - ⊥	219 - ⊥	235 - ы	251 - √
140 - M	156 - Ь	172 - м	188 - ⊥	204 - ⊥	220 - ⊥	236 - ь	252 - Ne
141 - H	157 - Э	173 - н	189 - ⊥	205 - =	221 - ⊥	237 - э	253 - Œ
142 - O	158 - Ю	174 - o	190 - ⊥	206 - ⊥	222 - ⊥	238 - ю	254 - █
143 - П	159 - Я	175 - п	191 - ⊥	207 - ⊥	223 - ⊥	239 - я	255 -
144 - P	160 - a	176 - ☼	192 - L	208 - ⊥	224 - p	240 - Ę	



Робота із символами реалізується за допомогою стандартних функцій.

Функції обробки символів визначають, чи являється цей символ цифрою, чи являється цей символ буквою та інше.

Для виклику функцій обробки символів необхідно підключити заголовний файл **<ctype.h>**.

Такі функції оперують із символами як з цілими числами (символу відповідає код).

Функції зчитування символів



Для зчитування одного символу з клавіатури передбачені функції ***getch()***, ***getche()*** , ***getchar()***

Функція ***getch()*** повертає символ, який зчитаний з консолі, але не виводить цей символ на екран.

Функція ***getche()*** повертає символ, який зчитаний з консолі та виводить цей символ на екран.

Функція ***getchar()*** повертає символ, який зчитаний з консолі та виводить цей символ на екран.

Завдання.



Розробити програму, що дозволяє зчитувати символ з клавіатури і друкувати його на екран у двох форматах: як символ (формат "%c") і як число (формат "%d").

Це відбувається до тих пір, поки символ (а точніше його ASCII код) не буде дорівнює 48, тобто поки не буде натиснута клавіша 0.

Рішення з getch()



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char chl;
6      do {
7          chl=getch();
8          printf ("You enter %c. ASCII code = %d\n", chl, (int)chl);
9      } while (chl != 48);
10
11     return 0;
12 }
13
```

A screenshot of a Windows command prompt window. The title bar reads "D:\Programms_Code\task\bin\Debug\task.exe". The window contains the following text:

```
You enter g. ASCII code = 103
You enter E. ASCII code = 69
You enter h. ASCII code = 104
You enter 0. ASCII code = 48

Process returned 0 (0x0)   execution time : 22.479 s
Press any key to continue.
-
```

Рішення з getchc()



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char chl;
6      do {
7          chl=getchc();
8          printf ("\nYou enter %c. ASCII code = %d\n", chl, (int)chl);
9      } while (chl != 48);
10
11     return 0;
12 }
13
```

A screenshot of a Windows command prompt window titled "D:\Programms_Code\task\bin\Debug\task.exe". The window shows the output of the program: "h", "u", "d", "6", and "0", each followed by "You enter [character]. ASCII code = [value]". The values are 104, 117, 100, 54, and 48 respectively. At the bottom, it says "Process returned 0 (0x0) execution time : 9.213 s" and "Press any key to continue.".

```
D:\Programms_Code\task\bin\Debug\task.exe
h
You enter h. ASCII code = 104
u
You enter u. ASCII code = 117
d
You enter d. ASCII code = 100
6
You enter 6. ASCII code = 54
0
You enter 0. ASCII code = 48

Process returned 0 (0x0)   execution time : 9.213 s
Press any key to continue.
```

Рішення з getchar()



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char chl;
7      do{
8          chl=getchar();
9          printf("\nYou enter %c. ASCII code = %d\n", chl, (int)chl);
10         }while (chl!=48);
11
12     return 0;
13 }
```

```
D:\Programms_Code\task3\bin\Debug\task3.exe
r
You enter r. ASCII code = 114

You enter
. ASCII code = 10
d
You enter d. ASCII code = 100

You enter
. ASCII code = 10
8
You enter 8. ASCII code = 56

You enter
. ASCII code = 10
0
You enter 0. ASCII code = 48

Process returned 0 (0x0)   execution time : 5.286 s
Press any key to continue.
```

Функції виведення символів



Для виведення одного символу передбачені функції

int putchar(int ch)

Ця функція не визначена стандартом ANSI C.

Функція ***putchar()*** виводить на екран символ, зазначений у ***ch***.

Ця функція виводить прямо на екран, а не в стандартний ***stdout***.

Тому її робота не супроводжується ні транслюванням, ні перенаправленням.



```
int putchar(int c);
```

Функція ***putchar*** ВИВОДИТЬ СИМВОЛ В стандартний потік виведення.

Ця функція аналогічна функції **putc**, при вказівці в якості потоку виведення стандартного потоку виводу (***stdout***)



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char chl;
6      do {
7          chl=getch();
8          printf ("\nYou enter ");
9          putchar(chl);
10
11         } while (chl != 48);
12
13         getch();
14         return 0;
15     }
16
```

D:\Programms_Code\task\bin\Debu

```
You enter 4
You enter t
You enter c
You enter s
You enter 0
```



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

int main(){
    char ch1;
    do {
        ch1=getch();
        printf ("\nYou enter ");
        putchar(ch1);

    } while (ch1 != 48);

    getch();
    return 0;
}
```



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char chl;
6      do {
7          chl=getch();
8          printf ("\nYou enter ");
9          putchar(chl);
10
11         // printf ("\nYou enter %c. ASCII code = %d\n", chl, (int)chl);
12     } while (chl != 48);
13     getch();
14
15     return 0;
16 }
17
```

D:\Programms_Code\task\bin\Debug\task.exe

```
You enter r
You enter g
You enter x
You enter 0
Process returned 0 (0x0)   execution time : 9.993 s
Press any key to continue.
```



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

int main(){
    char ch1;
    do {
        ch1=getch();
        printf ("\nYou enter ");
        putchar(ch1);

        // printf ("\nYou enter %c. ASCII code = %d\n", ch1, (int)ch1);
    } while (ch1 != 48);
    getch();

    return 0;
}
```

Функції для роботи з символами



isalnum(c)	цифра або літера
isalpha(c)	літера
iscntrl(c)	керуючий символ
isdigit(c)	десяткова цифра
isgraph(c)	друкований символ, крім пробілу
islower(c)	літера нижнього регістру
isupper(c)	літера верхнього регістру

Функції для роботи з символами



isprint(c)	друкований символ, також і пробіл
ispunct(c)	друкований символ крім пробілу, літери або цифри
isspace(c)	пробіл, змінення сторінки, нова сторінка, повернення каретки, табуляція, вертикальна табуляція
isxdigit(c)	шіснадцятерична цифра
int tolower(int c)	переводить c на нижній регістр
int toupper(int c)	переводить c на верхній регістр

Приклад



Визначити, чи є символ літерою чи цифрою

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 int main() {
6     char c;
7
8     printf("Enter character ");
9     scanf("%c", &c);
10
11     if (isdigit(c)) {
12         printf("digit\n");
13     } else if (isalpha(c)) {
14         printf("alpha\n");
15     } else {
16         printf("error.\n");
17     }
18     return 0;
19 }
```

```
Enter character 66
digit
```

```
Enter character f
alpha
```

```
Enter character &
error.
```

Лістинг програми



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

```
int main() {
    char c;
```

```
    printf("Enter character ");
    scanf("%c", &c);
```

```
    if (isdigit(c)) {
        printf("digit\n");
    } else if (isalpha(c)) {
        printf("alpha\n");
    } else {
        printf("error.\n");
    }
    return 0;
}
```



```
1  #include <stdio.h>
2  #include <ctype.h>
3  int main() {
4      char c;
5      do {
6          printf("Enter character: ");
7          c = getche();
8          printf("character: %c\n", c);
9          printf("code: %d\n", c);
10
11         if (isalnum(c)) {
12             printf("character number-letter\n");
13         } else if (ispunct(c)) {
14             printf("character punctuation mark\n");
15         } else if (iscntrl(c)) {
16             printf("character control\n");
17         } else if (isspace(c)) {
18             printf("character space\n");
19         } else {
20             printf("error.\n");
21         }
22         printf("\n");
23     } while (c != 27);
24
25     return 0;
26 }
27
```

D:\Programms_Code\task1\bin\Debug\task1.exe

```
Enter character: 8character: 8
code: 56
character number-letter
```

```
Enter character: kcharacter: k
code: 107
character number-letter
```

```
Enter character: +character: +
code: 43
character punctuation mark
```

```
Enter character: \character: \
code: 92
character punctuation mark
```

```
Enter character: _
```

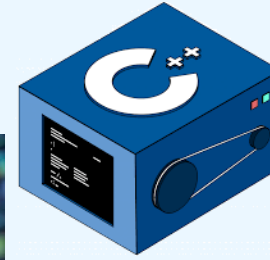
Лістинг програми



```
#include <stdio.h>
#include <ctype.h>
int main() {
    char c;
    do {
        printf("Enter character: ");
        c = getche();
        printf("character: %c\n", c);
        printf("code: %d\n", c);
```

```
        if (isalnum(c)) {
            printf("character number-letter\n");
        } else if (ispunct(c)) {
            printf("character punctuation mark\n");
        } else if (iscntrl(c)) {
            printf("character control\n");
        } else if (isspace(c)) {
            printf("character space\n");
        } else {
            printf("erorr.\n");
        }
        printf("\n");
    } while (c != 27);

    return 0;
}
```



```
each: function(e, n) {
  var r, i = 0;
  m = e.length;
  n = n || 0;
  if (n) {
    if (n < 0) {
      for (; n > -1; i++)
        if (r = t.apply(e[i], n), r === !1) break;
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break;
    } else if (e) {
      for (; o > 1; i++)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    } else
      for (i in e)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    return e;
  };
  trim: b && !b.call("u0eff\u0000") ? function(e) {
    return null == e ? "" : b.call(e);
  } : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
  };
  isArray: function(e, t) {
    var n = t || [];
    return null != e && (N(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : b.call(n, e));
  };
  isArray: function(e, t, n) {
    var r;
    if (n)
      if (n) return b.call(t, n);
      for (r = 0; r < e.length; r++)
        if (r > 0 && n > n ? Math.max(0, r + n) : n < 0; r > n; r++)
          if (n < 0 && r[n] === n) return 0;
  }
}
```

Дякую за увагу!!!