ВикипедиЯ

ctype.h

Материал из Википедии — свободной энциклопедии

ctype.h — заголовочный файл стандартной библиотеки языка программирования C, содержащий объявления функций для классификации и преобразования отдельных символов.

Содержание

Функции

Все перечисленные ниже функции имеют прототипы следующего вида:

`int` *`имя_функции`*`( int );`

Все эти функции принимают в качестве аргумента `int`-эквивалент символа и возвращают значение типа `int`, которое может представлять или другой символ (в случае функций преобразования), или логическое значение: 0 означает «Ложь», а отличное от нуля значение — «Истина» (это относится к функциям классификации).

Поведение этих функций зависит от текущей кодировки.[1]

Результат применения этих функций к символам с кодами, принадлежащими расширенной кодировке времени выполнения (англ. *extended execution characterset*) зависит от платформы и локализации.

Функции классификации

```c
#include <ctype.h>
int isalnum(int c); //Если аргумент функции является либо буквой, либо цифрой, она возвращает ненулевое значение.
int isalpha(int c); //Возвращает ненулевое значение, если её аргумент является буквой, в противном случае возвращается нуль.
int isblank(int c); //Возвращает true, если c - пробел или горизонтальная табуляция (C99).
int iscntrl(int c); //Возвращает true, если c - управляющий символ, такой как <Ctrl+B>.
int isdigit(int c); //Возвращает ненулевое значение, если её аргумент является десятичной цифрой, в противном случае возвращается нуль.
int isgraph(int c); //Возвращает true, если c - печатаемый символ, отличный от пробела.
int islower(int c); //Возвращает true, если c - символ нижнего регистра.
int isprint(int c); //Возвращает true, если c – печатаемый символ.
int ispunct(int c); //Возвращает true, если c - знак препинания (любой печатаемый символ, отличный от пробела или алфавитно-цифрового символа).
```

```
int isspace(int c); //Возвращает true, если c — пробельный символ: пробел, новая строка,перевод страницы, возврат каретки, вертикальная табуляция, горизонтальная табуляция или, возможно, другой символ, определяемый реализацией
int isupper(int c); //Возвращает true, если c - символ верхнего регистра.
int isxdigit(int c); //Возвращает true, если c — шестнадцатеричная цифра.
```

Эти функции проверяют, является ли аргумент буквой или цифрой, пробелом или табуляцией, управляющим символом, десятичным числом, печатным символом (кроме пробела), символом в нижнем регистре, печатным символом (в том числе пробелом), пробелом, символом в верхнем регистре или шестнадцатиричным числом.

Функции преобразования

```
#include <ctype.h>
int toupper(int c);   // переводит буквы нижнего регистра в верхний регистр
int tolower(int c);   // переводит буквы верхнего регистра в нижний регистр
```

Функции преобразуют символ c в нижний или верхний регистр, если это возможно. В противном случае они возвращают неизменённое значение.[1]

Реализация функций

В большинстве библиотек языка C подпрограммы классификации используют статические таблицы поиска вместо макросов или функций. Например, создается массив из 256 восьмибитовых целых чисел, где каждый бит означает определенное свойство символа (является цифрой, буквой и т. д.). Если бит 1 показывает, цифра данный символ или нет, то код мог бы быть записан так:

```
#define isdigit(x) (TABLE[x] & 1)
```

Ранние версии Linux использовали потенциально опасный код, подобный следующему:

```
#define isdigit(x) ((x) >= '0' && (x) <= '9')
```

Это могло вызвать проблемы, если в качестве аргумента использовался, например, вызов функции. В таком случае сразу не было видно, что функция вызывается дважды.

Неправильное использование

> *Во всех случаях аргумент имеет тип `int`, причем его значение должно принадлежать диапазону значений типа `unsigned char`, или же быть равным значению макроконстанты EOF. Если аргумент имеет любое другое значение, поведение функций не определено.*

Тип `char` в зависимости от реализации может быть как знаковым, так и беззнаковым. Если тип `char` является знаковым, то неявное преобразование в `int` может привести к появлению отрицательных чисел, что вызовет неопределенное поведение функции. Чаще всего происходит выход за пределы таблицы поиска и аварийное завершение программы.

```
//Этот код может работать некорректно.
char test = 'b';

toupper( 'a' );
int res = ispunct( test );
```

Чтобы избежать этого, используя `char`-аргументы, сперва явно преобразуйте их в `int`.

`int`-значения, возвращаемые функциями `getchar`, `getc`, `fgetc` обязательно принадлежат диапазону значений типа `unsigned char` (или равны EOF), поэтому здесь преобразование не требуется.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <locale.h>
#include <ctype.h>
#include <string.h>

int main()
{
    setlocale(LC_ALL, "RUS");
    char s[100];
    int i, ind, n; //Исходные данные
    char ch[4];

    printf("Введите A, B и C, без пробелов:");
    gets_s(ch);

    printf("Введите текст:\n");
    gets_s(s); //gets(функция, входящая в Стандартную библиотеку языка Си, объявляемая в заголовочном
файле stdio.h , которая считывает строку стандартного ввода и помещает её в буфер, созданный вызывающей
функцией).

    n = strlen(s); //strlen(видит начало строки  и начинает сначала считать количество символов (байтов,
отводимых под каждый
    if (islower(ch[1]) || islower(ch[2])) //принимает в качестве аргумента один символ (букву) и
возвращает ненулевое целое значение в том случае, если буква является строчной, и нулевое, если буква
является заглавной

    ch[1] = ch[1] - 32;
    ch[2] = ch[2] - 32;

    for (i = 0; i< n; i++)
    {
        if (s[i] == ch[0])
        {
            ind = i;
            if (isdigit(s[i - 1])) //isdigit (Функция isdigit проверяет аргумент, передаваемый через
параметр character, является ли он десятичной цифрой.
            {
                break;
            }
            else
            {
                s[i] = ch[1];
            }
        }
    }

    for (i = n + 1; i>ind; i--)
    {
        s[i] = s[i - 1];
    }

    s[ind + 1] = ch[2];
    puts(s);
    _getch();
}
```

Примеры использования

```
//Следующий код считывает символ и преобразует его в
//число типа int, если введена цифра.
#include <stdio.h>
```

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

int main (void)
{
  int c = fgetc(stdin);
  if (c != EOF)
  {
     if ( isdigit(c) )
         printf( "You have entered a number %i\n", atoi((char*)&c)  );
     else
         printf( "It is not a number!\n" );
  }
  return 0;
}
```

Ссылки

- ctype.h (http://www.opengroup.org/onlinepubs/9699919799/basedefs/ctype.h.html) — основные определения, The Single UNIX® Specification, выпуск 7 от The Open Group (англ.)
- ctype.h на C++ Reference (http://www.cplusplus.com/reference/clibrary/cctype/)

Примечания

1. ISO/IEC 9899:1999 (http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1124.pdf). Дата обращения: 31 июля 2011. Архивировано (https://www.webcitation.org/618HUIDTw?url=http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1124.pdf) 22 августа 2011 года.

Источник — https://ru.wikipedia.org/w/index.php?title=Ctype.h&oldid=122126492
