

# **Компьютерные технологии в научных исследованиях и образовании**

Юрий Анатольевич Богоявленский, заведующий кафедрой Информатики и математического обеспечения, к.т.н., доцент, [ybgv](mailto:ybgv)

## **Ввод и вывод**

### **Вывод на терминал**

#### **Функции `disp` и `format`**

#### **Вызовы функции `format`**

#### **Примеры использования функции `format`.**

#### **Постраничный вывод**

### **Ввод с терминала**

Поддерживается несколько способов чтения и записи данных как в ответ на приглашение интерпретатора `octave`, так и с использованием файлов.

Простые функции ввода/вывода легко применять, но их возможности ограничены. Реализован также полноценный набор этих функций на базе библиотеки `libc`, входящей в стандарт языка C.

## **Простые функции ввода/вывода**

### **Вывод на терминал**

Вызов функция `terminal_size` () без параметров формирует вектор из двух элементов — установленных для терминала текущих значений количества строк и символов в них. Вызов этой функции с фактическим параметром — таким массивом, содержащим необходимые значения этих количеств приведет к их установке с выводом их предыдущих значений. Если для редактирования командной строки используется библиотека `readline`, то этой функцией пользоваться не требуется.

После ввода имени переменной или выражения не заканчивающихся на символ точка с запятой octave преобразует внутреннее представления их значения в символьную строку, выводит имя переменной (символы ans для выражения), знак равенства, символьную строку, представляющую значение и переходит на новую строку. Способ преобразования можно задать функцией `format`.

## Функции `disp` и `format`

Если нужно вывести только значение то используется функция `disp (x)`, которая преобразует значение `x` в символьную строку также используя способы преобразования, заданные функцией `format`. Символьная строка не выводится, если ее значение присваивается переменной, например так:

```
str = disp (x)
```

Функция `format`:

- при вызове с входными параметрами задает способы преобразования внутреннего представления числовых значений при выводе, а также некоторые режимы вывода;
- при вызове с выходными параметрами позволяет получить установленные в системе текущие данные об этих способах и режимах.

Входные фактические параметры функции можно указывать без заключения в скобки, разделяя их пробелами. Они делятся на три группы: F, D и C, причем значения в каждой группе фиксированы. Можно вызывать функцию `format` без параметров или указывать любое их число, причем порядок элементов групп не имеет значения. Подчеркнем, что если указано более чем одно значение параметра из одной группы, то будет действовать самое правое из них.

Рассмотри группы фактических параметров.

Группа F.

Элементы этой группы описывают способы преобразования внутреннего представления их допустимые значения приведены в следующей таблице.

Параметр	Описание
<i>Только Для значений с плавающей точкой</i>	
short	<b>По умолчанию.</b> Фиксированная точка, до 5 значащих цифр.
long	Фиксированная точка, 16 значащих цифр. Как и для формата short, octave переключится на соответствующий экспоненциальный формат "e", если не возможно правильно отформатировать матрицу, используя текущий формат.
short e long e	Экспоненциальный формат в виде мантиссы с показателем степени 10. Мантисса имеет 5 или 16 значащих цифр. Например число pi выведется как 3.1416e+00 или 3.141592653589793E+00
short g long g	Выполняется рациональный выбор между форматом с фиксированной точкой и экспоненциальным в зависимости от величины числа. Например, с форматом 'short g', выражение pi.^ [2; 4; 8; 16; 32] выведется как:  ans =9.8696 97.409 9488.5 9.0032e+07 8.1058e+15
short eng long eng	Идентичен short e или long e, но выводит значение в инженерном формате, где порядок кратен 3. Например, в формате short eng значение 10 * pi выведется как 31.416e+00.

free none	<p>Вывод в свободном формате, без выравнивания столбцов матриц по положению десятичной точки. В общем случае будет представление с 6 значащими цифрами с подавлением незначащих нулей. Комплексные числа выводятся как числовые пары — (0.60419, 0.60709), а не как 0.60419 + 0.60709i.</p>
<p><i>Для значений как типов с плавающей точкой, так и целого типа</i></p>	
"+" "+" "chars" plus plus "chars"	<p>Выводится символ "+" для элементов матрицы, больших нуля, символ "-" для элементов, меньших нуля, и пробел для нулевых элементов. Формат может быть полезен для изучения структуры разрежения большой матрицы.</p> <p>Необязательный аргумент chars задает список из 3 символов, который будет использоваться для вывода значений больше нуля, меньше нуля и равных нулю.</p> <p>Например, с форматом "+" "+-.", матрица [1, 0, -1; -1, 0, 1] введется как:</p> <pre>ans = +.- - .+</pre>
bank	<p>Вывод в формате, принятом для денежных сумм с двумя цифрами справа от десятичной точки. Мнимая часть не отображается.</p>
native-hex	<p>Выводит шестнадцатеричное представление чисел в том виде, в каком они хранятся в памяти. Например, для 8-байтовых действительных значений в формате IEEE с младшим адресом наименьшего значащего байта, значение pi будет выведено как 400921fb54442d18.</p>
hex	<p>Аналогичен native-hex, но старший значащий байт всегда выводится первым.</p>
native-bit	<p>Выведит битовое представление значения в памяти.</p>

	<p>Например, для 8-байтовых действительных значений в формате IEEE с младшим адресом наименьшего значащего байта, значение <math>\pi</math> (разбитое на две 32-битовые строки) будет выведено как:</p> <pre>010000000000010010010000111111011 01010100010001000010110100011000</pre>
bit	Аналогичен native-bit, но старший значащий байт всегда выводится первым.
rat	Выводится рациональное приближение, т.е. значение аппроксимируются как отношение малых целых чисел. Например, в формате число $\pi$ выведется как 355/113.

В группе D всего два значения входных фактических параметров, которые задают стиль вывода массивов:

- compact — не выводятся пустые строки вокруг меток с номерами столбцов и между матрицами, что дает более компактный вывод с большим количеством данных на странице;
- loose (**по умолчанию**) — указанные выше пустые строки выводятся, что дает более удобочитаемый результат с меньшим количеством данных на странице.

Наконец в группе C также два значения входных фактических параметров, задающие вывод строчных или прописных букв в экспоненциальной (научной форме) чисел и в шестнадцатеричном предоставлении:

- lowercase (**по умолчанию**)— строчные буквы;
- uppercase — прописные буквы.

## Вызовы функции format

`format` без параметров — устанавливает значения по умолчанию для параметров всех трех групп - `short`, `lowercase`, и `loose`

`format (arg-list)` — вызов с входными параметрами , устанавливает указанные значения, например три эквивалентных вызова:

```
format long uppercase compact
```

```
format uppercase compact long
```

```
format compact long uppercase
```

Вызов без входных параметров, но с указанием от одного до трех выходных, дает в виде заключенных в кавычки текстовых строк (тип `sq_string`) текущие значения параметров. Если выходных параметров 2 или 3, то нужно задавать их в виде вектора, элементам которого `format` присвоит значения типа `sq_string` текстовых строк. При этом значения присваиваются им в порядке F, D, C групп параметров, как это показано на примере ниже.

```
ybgv@ybgv-home:~> octave -q -p ~/MyOct
octave:1> newwhos
octave:2>
octave:2> # один выходной параметр
octave:2>
octave:2> a=format
a = short
octave:3> # два выходных параметра
octave:3>
octave:3> [a,b]=format
a = short
b = loose
octave:4> # три выходных параметра
octave:4>
```

```
octave:4> [a,b,c]=format
a = short
b = loose
c = lowercase
octave:5> whos
Variables visible from the current scope:
```

```
variables in scope: top scope
```

Attr	Name	Size	Type	Bytes	Class
====	====	====	====	=====	=====
	a	1x5	sq_string	5	char
	ans	1x10	sq_string	10	char
	b	1x5	sq_string	5	char
	c	1x9	sq_string	9	char

```
Total is 29 elements using 29 bytes
```

```
octave:6>
```

## Примеры использования функции format.

```
bgv@ybgv-home:~> octave -q -p ~/MyOct
octave:1> frmt
+ echo on
+
+ q = -345.7897915
q = -345.79
+
+ format
```

```

+
+ [type, dens,ulcase] = format
type = short
dens = loose
ulcase = lowercase
+
+ # задание матрицы двойной точности
+
+ B=repelem ([-1, 0; 0, 1], 3, 3)
B =

-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1

+
+ B(1,1)=pi;B(2,2)=e;B(3,3)=0.666666;
+ B(4,4) = 10.^2; B(5,5) = 10.^4;B(6,6) = 10.^6;
+
+ # задание целой матрицы
+
+ C=int32(repelem ([-1, 0; 0, 1], 3, 3))
C =

-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1

```



```

    0    0    0    1    1    1

+
+ format short; [type, dens,ulcase] = format
type = short
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.79
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
Columns 1 through 5:

    3.1416e+00   -1.0000e+00   -1.0000e+00           0           0
   -1.0000e+00    2.7183e+00   -1.0000e+00           0           0
   -1.0000e+00   -1.0000e+00    6.6667e-01           0           0
           0           0           0    1.0000e+02    1.0000e+00
           0           0           0    1.0000e+00    1.0000e+04
           0           0           0    1.0000e+00    1.0000e+00

Column 6:

           0
           0
           0
    1.0000e+00
    1.0000e+00
    1.0000e+06
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
   -1   -1   -1    0    0    0

```

```

-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
-- конец C
+
+ format short g;[type, dens,ulcase] = format
type = shortg
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.79
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
Columns 1 through 5:

    3.1416         -1         -1          0          0
    -1         2.7183         -1          0          0
    -1         -1         0.66667          0          0
    0          0          0         100          1
    0          0          0          1         10000
    0          0          0          1          1

Column 6:

    0
    0
    0
    1
    1
1e+06

```

```

-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
  -1  -1  -1  0  0  0
  -1  -1  -1  0  0  0
  -1  -1  -1  0  0  0
   0   0   0  1  1  1
   0   0   0  1  1  1
   0   0   0  1  1  1
-- конец C
+
+ format short eng; [type, dens,ulcase] = format
type = shorteng
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.7898e+000
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
Columns 1 through 4:

    3.1416e+00    -1.0000e+00    -1.0000e+00         0
   -1.0000e+00     2.7183e+00    -1.0000e+00         0
   -1.0000e+00    -1.0000e+00    666.6660e-03         0
                0                0                0    100.0000e+00
                0                0                0     1.0000e+00
                0                0                0     1.0000e+00

Columns 5 and 6:

                0                0

```

```

0 0
0 0
1.0000e+00 1.0000e+00
10.0000e+03 1.0000e+00
1.0000e+00 1.0000e+06
-- конец В
+ disp("Матрица С, тип int32");disp(C);disp("-- конец С")
Матрица С, тип int32
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
-- конец С
+
+ format long; [type, dens,ulcase] = format
type = long
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.7897915000000
+ disp("Матрица В, тип double");disp(B);disp("-- конец В")
Матрица В, тип double
Columns 1 through 3:

3.141592653589793e+00 -1.000000000000000e+00 -1.000000000000000e+00
-1.000000000000000e+00 2.718281828459045e+00 -1.000000000000000e+00
-1.000000000000000e+00 -1.000000000000000e+00 6.666660000000000e-01
0 0 0
0 0 0
0 0 0

```

Columns 4 through 6:

```
          0          0          0
          0          0          0
          0          0          0
1.0000000000000000e+02  1.0000000000000000e+00  1.0000000000000000e+00
1.0000000000000000e+00  1.0000000000000000e+04  1.0000000000000000e+00
1.0000000000000000e+00  1.0000000000000000e+00  1.0000000000000000e+06
```

-- конец B

```
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
```

Матрица C, тип int32

```
-1 -1 -1  0  0  0
-1 -1 -1  0  0  0
-1 -1 -1  0  0  0
 0  0  0  1  1  1
 0  0  0  1  1  1
 0  0  0  1  1  1
```

-- конец C

+

```
+ format long e; [type, dens,ulcase] = format
```

```
type = longe
```

```
dens = loose
```

```
ulcase = lowercase
```

```
+ disp("q = ");disp(q);
```

```
q =
```

```
-3.457897915000000e+02
```

```
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
```

Матрица B, тип double

Columns 1 through 3:

```
 3.141592653589793e+00 -1.0000000000000000e+00 -1.0000000000000000e+00
-1.0000000000000000e+00  2.718281828459045e+00 -1.0000000000000000e+00
-1.0000000000000000e+00 -1.0000000000000000e+00  6.666660000000000e-01
```

```

0 0 0
0 0 0
0 0 0

Columns 4 through 6:

0 0 0
0 0 0
0 0 0
1.0000000000000000e+02 1.0000000000000000e+00 1.0000000000000000e+00
1.0000000000000000e+00 1.0000000000000000e+04 1.0000000000000000e+00
1.0000000000000000e+00 1.0000000000000000e+00 1.0000000000000000e+06
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
-- конец C
+
+ format long g; [type, dens,ulcase] = format
type = longg
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.7897915
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
Columns 1 through 3:

```

```

3.141592653589793          -1          -1
          -1          2.718281828459045          -1
          -1          -1          0.666666
          0          0          0
          0          0          0
          0          0          0

Columns 4 through 6:

          0          0          0
          0          0          0
          0          0          0
        100          1          1
          1          10000          1
          1          1          1000000

-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
-1 -1 -1  0  0  0
-1 -1 -1  0  0  0
-1 -1 -1  0  0  0
 0  0  0  1  1  1
 0  0  0  1  1  1
 0  0  0  1  1  1

-- конец C
+
+ format long eng; [type, dens,ulcase] = format
type = longeng
dens = loose
ulcase = lowercase
+ disp("q = ");disp(q);
q =
-345.789791499999978e+000
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")

```

Матрица B, тип double

Columns 1 and 2:

3.141592653589793e+00	-1.0000000000000000e+00
-1.0000000000000000e+00	2.718281828459045e+00
-1.0000000000000000e+00	-1.0000000000000000e+00
0	0
0	0
0	0

Columns 3 and 4:

-1.0000000000000000e+00	0
-1.0000000000000000e+00	0
666.665999999999940e-03	0
0	100.0000000000000000e+00
0	1.0000000000000000e+00
0	1.0000000000000000e+00

Columns 5 and 6:

0	0
0	0
0	0
1.0000000000000000e+00	1.0000000000000000e+00
10.0000000000000000e+03	1.0000000000000000e+00
1.0000000000000000e+00	1.0000000000000000e+06

-- конец B

+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")

Матрица C, тип int32

-1	-1	-1	0	0	0
-1	-1	-1	0	0	0





Columns 5 and 6:

```

0
0
0
1.0000000000000000E+00 1.0000000000000000E+00
10.0000000000000000E+03 1.0000000000000000E+00
1.0000000000000000E+00 1.0000000000000000E+06
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
```

Матрица C, тип int32

```

-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
```

-- конец C

+

```
+ format long eng uppercase compact; [type, dens,ulcase] =
format
```

```
type = longeng
```

```
dens = compact
```

```
ulcase = uppercase
```

```
+ disp("q = ");disp(q);
```

```
q =
```

```
-345.789791499999978E+000
```

```
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
```

Матрица B, тип double

Columns 1 and 2:

```
3.141592653589793E+00 -1.0000000000000000E+00
```

```
-1.0000000000000000E+00    2.718281828459045E+00
-1.0000000000000000E+00    -1.0000000000000000E+00
                                0                                0
                                0                                0
                                0                                0
```

Columns 3 and 4:

```
-1.0000000000000000E+00    0
-1.0000000000000000E+00    0
666.665999999999940E-03    0
                                0    100.000000000000000E+00
                                0    1.0000000000000000E+00
                                0    1.0000000000000000E+00
```

Columns 5 and 6:

```
                                0                                0
                                0                                0
                                0                                0
1.0000000000000000E+00    1.0000000000000000E+00
10.0000000000000000E+03    1.0000000000000000E+00
1.0000000000000000E+00    1.0000000000000000E+06
```

```
-- конец B
```

```
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
```

```
Матрица C, тип int32
```

```
-1  -1  -1  0  0  0
-1  -1  -1  0  0  0
-1  -1  -1  0  0  0
 0   0   0  1  1  1
 0   0   0  1  1  1
 0   0   0  1  1  1
```

```
-- конец C
```

```
+
```

```
+ format free; [type, dens,ulcase] = format
```

```
type = free
```

```

dens = compact
ulcase = uppercase
+ disp("q = ");disp(q);
q =
-345.79
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
 3.14159 -1 -1 0 0 0
-1 2.71828 -1 0 0 0
-1 -1 0.666666 0 0 0
0 0 0 100 1 1
0 0 0 1 10000 1
0 0 0 1 1 1e+06
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1

-- конец C
+
+ format "+"; [type, dens,ulcase] = format
type = +
dens = compact
ulcase = uppercase
+ disp("q = ");disp(q);
q =
-

```

```

+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
+--
-+-
--+
    +++
    +++
    +++
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
---
---
---
    +++
    +++
    +++
-- конец C
+
+ format "+" "pmz"; [type, dens,ulcase] = format
type = +pmz
dens = compact
ulcase = uppercase
+ disp("q = ");disp(q);
q =
m
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
pmmzzz
mpmzzz
mmpzzz
zzzppp

```

```

zzzppp
zzzppp
-- конец В
+ disp("Матрица С, тип int32");disp(C);disp("-- конец С")
Матрица С, тип int32
mmmzzz
mmmzzz
mmmzzz
zzzppp
zzzppp
zzzppp
-- конец С
+
+ format native-hex; [type, dens,ulcase] = format
type = native-hex
dens = compact
ulcase = uppercase
+ disp("q = ");disp(q);
q =
8A7269FCA29C75C0
+ disp("Матрица В, тип double");disp(B);disp("-- конец В")
Матрица В, тип double
Columns 1 through 4:
      182D4454FB210940      000000000000F0BF      000000000000F0BF
0000000000000000
      000000000000F0BF      6957148B0ABF0540      000000000000F0BF
0000000000000000
      000000000000F0BF      000000000000F0BF      465D6BEF5355E53F
0000000000000000
      0000000000000000      0000000000000000      0000000000000000
00000000000005940

```

```

00000000000000000000 00000000000000000000 00000000000000000000
00000000000000F03F
00000000000000000000 00000000000000000000 00000000000000000000
00000000000000F03F
Columns 5 and 6:
00000000000000000000 00000000000000000000
00000000000000000000 00000000000000000000
00000000000000000000 00000000000000000000
00000000000000F03F 00000000000000F03F
0000000000008C340 000000000000F03F
000000000000F03F 00000000080842E41
-- конец B
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
FFFFFFFF FFFFFFFFF FFFFFFFFF 00000000 00000000 00000000
FFFFFFFF FFFFFFFFF FFFFFFFFF 00000000 00000000 00000000
FFFFFFFF FFFFFFFFF FFFFFFFFF 00000000 00000000 00000000
00000000 00000000 00000000 01000000 01000000 01000000
00000000 00000000 00000000 01000000 01000000 01000000
00000000 00000000 00000000 01000000 01000000 01000000
-- конец C
+
+ format rat; [type, dens,ulcase] = format
type = rat
dens = compact
ulcase = uppercase
+ disp("q = ");disp(q);
q =
-54289/157
+ disp("Матрица B, тип double");disp(B);disp("-- конец B")
Матрица B, тип double
355/113 -1 -1 0 0 0

```

```

-1 1457/536 -1 0 0 0
-1 -1 2/3 0 0 0
0 0 0 100 1 1
0 0 0 1 10000 1
0 0 0 1 1 *
```

-- конец B

```
+ disp("Матрица C, тип int32");disp(C);disp("-- конец C")
Матрица C, тип int32
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
-1 -1 -1 0 0 0
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
```

-- конец C

```
+
+
octave:2>
```

## Постраничный вывод

Команды `octave` могут генерировать вывод текста большого объема, не помещающегося на экране терминала. В таких случаях для просмотра текста можно использовать программы постраничного вывода, позволяющие перемещаться по тексту с помощью клавиш стрелок и управления страницами. Функция `more` включает/выключает этот вид вывода.

`more on` — включить;

`more off` — выключить;

`more` — переключить на противоположный текущему.

Функция `page_screen_output` выводит 1, если этот вид вывода включен и 0 — если выключен.



Мы не рассматриваем данные в [Док, с. 273 — 274]) другие функции управления постраничным выводом.

## **Ввод с терминала**

В `octave` есть три функции обеспечивающие ввод данных пользователя с подсказкой. Функция может быть задана в двух формах:

```
input (prompt)
```

```
input (prompt, "s")
```

Ожидание ввода с клавиатуры происходит после вывода на экран строки `prompt`. В первом случае функция возвращает значение, получаемое при вычислении введенной строки, которая рассматривается как допустимое выражение `octave`.

Если присутствует второй параметр — `"s"`, то функция возвращает значение введенной последовательность символов, трактуя ее как текстовую строку, а не как выражение `octave`.

Если включен постраничный вывод, то могут быть ожидающие вывода данные. В этом случае рекомендуется перед вызовом функции `input` форсировать вывод этих данных вызовом функции `fflush (stdout)`.

Функция `menu` позволяет организовать выбор из нескольких вариантов и задается в виде:

```
menu (title, opt1, . . . )
```

```
menu (title, {opt1, . . . })
```

где:

`title` — текстовая строка с заголовком меню;

`opt1, . . .` — несколько текстовых строк имен вариантов выбора.

Если графический интерфейс запущен, меню отображается графически с помощью функции `listdlg`. В противном случае заголовок и параметры меню выводятся на терминал. Функция возвращает значение номера

варианта, выбранного пользователем. Если он закрывает диалоговое окно или делает неверный выбор, то возвращается значение 0.

Функция `yes_or_no ("prompt")` имеет один параметр — текстовую строку `"prompt"` и выводит на экран текст `"prompt"(yes or no)`. Пользователь в ответ должен набрать текст `"yes"` (возвращается значение `true`) или `"no"` (возвращается значение `false`).