

федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Московский институт электронной техники»

Направление подготовки 09.03.04 «Программная инженерия»
Дисциплина «Электротехника»

Лабораторная РАБОТА №4
Ввод-вывод при помощи libс

Работу выполнили студенты группы ПИН-24 Баранов Д.А. и Демочкина А.В.
Работу проверил ассистент Института СПИНТех Фомин Р.А.

Цель работы: научиться использовать функции ввода-вывода `libc`.

Для выполнения заданий выбран онлайн компилятор <https://godbolt.org/>.

ОС и разрядность ОС: GNU/Linux 64

Компилятор: Compiler Explorer x86-64 gcc 11.2

Архитектура: x86-64

Задание 1.

Разработайте программу, выводящую на стандартный вывод группу, номер и состав команды при помощи функции `puts()` библиотеки `libc`.

```
#include <stdio.h>
using namespace std;

int main() {
    char group[] = "ПИН";
    char number[] = "24";
    char teamComposition[] = "Баранов Дмитрий и Демочкина Александра";

    puts(group);
    puts(number);
    puts(teamComposition);
}
```

Код:

```
#include <stdio.h>
using namespace std;

int main() {
    char group[] = "ПИН";
    char number[] = "24";
    char teamComposition[] = "Баранов Дмитрий и Демочкина Александра";

    puts(group);
    puts(number);
    puts(teamComposition);
}
```

Вывод:

```
ПИН
24
Баранов Дмитрий и Демочкина Александра
```

Задание 2.

Вариант 2

2	$x = e$ с максимально возможной для типа точностью
---	--

Разработайте программу на языке C/C++, создающую, инициализирующую одинаковыми значениями и выводящую на экран при помощи функции `libc printf()` массивы из $N = 5$ чисел:

- M_s из 16-битных целых чисел (0xC0DE);
- M_l из 32-битных целых чисел (0x DEAD BEEF);
- M_q из 64-битных целых чисел (0x BECA CA01 FFED C0CA);
- M_{fs} из 32-битных чисел с плавающей запятой (значение x по варианту);
- M_{fl} из 64-битных чисел с плавающей запятой (значение x по варианту).

Выведите каждый из целочисленных массивов как в знаковом десятичном (d), так и в шестнадцатеричном (X) виде, чтобы убедиться, что короткие значения не расширены до 32 бит, а длинные — не усечены.

```
#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

template <typename T>
T show(T arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%X ", arr[i]);
    printf("\n\n");
    return 0;
}

int main() {
    unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
    unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF};
    unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA};
    float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
    double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
    show(ms);
    show(ml);
    show(mq);
}

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

template <typename T>
T show(T arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%X ", arr[i]);
    printf("\n\n");
    return 0;
}

int main() {
```

```

unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF,
                    0xDEADBEEF};
unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                    0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                    0xBECACA01FFEDC0CA};
float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
show(ms);
show(ml);
show(mq);
}

```

Вывод массива в знаковом десятичном виде:

49374 49374 49374 49374 49374

Вывод массива в шестнадцатеричном виде:

CODE CODE CODE CODE CODE

Вывод массива в знаковом десятичном виде:

-559038737 -559038737 -559038737 -559038737 -559038737

Вывод массива в шестнадцатеричном виде:

DEADBEEF DEADBEEF DEADBEEF DEADBEEF DEADBEEF

Вывод массива в знаковом десятичном виде:

-1195830 -1195830 -1195830 -1195830 -1195830

Вывод массива в шестнадцатеричном виде:

FFEDC0CA FFEDC0CA FFEDC0CA FFEDC0CA FFEDC0CA

Итог: короткие 16-битные значения были расширены до 32 бит, длинные 64-битные значения были наоборот усечены до 32 бит.

При выводе значения “-4” формата short функция printf() преобразовывала это значение в формат int и выводила 32 бита. Аналогично для формата long, только происходило усечение первой половины числа, т.е. происходил вывод только 32 бит.

Задание 3.

Для каждого массива M введите с клавиатуры новое значение элемента $M[i]$, $i = 2$ при помощи функции libc `scanf()`. Проанализировав возвращённое `scanf()` значение, определите корректность ввода. Выведите массивы на экран снова, убедитесь, что элемент $M[i]$ приобрёл ожидаемое значение, а другие элементы массива не изменились.

```

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

void showShort(unsigned short arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%hd ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%hX ", arr[i]);
    printf("\n\n");
}

void showInt(unsigned int arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%X ", arr[i]);
    printf("\n\n");
}

void showLong(unsigned long arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%ld ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%lX ", arr[i]);
    printf("\n\n");
}

void showFloat(float arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%f ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%a ", arr[i]);
    printf("\n\n");
}

void showDouble(double arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%lf ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%la ", arr[i]);
    printf("\n\n");
}

int main() {
    unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
    unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF,
        0xDEADBEEF};
    unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
        0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
        0xBECACA01FFEDC0CA};
    float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
    double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};

    int i = 2;
    scanf("%hu", &ms[i]); //считывание беззнакового целого 16 бит
    scanf("%du", &ml[i]); //считывание беззнакового целого 32 бита
    scanf("%lu", &mq[i]); //считывание беззнакового целого 64 бита
    scanf("%f", &mfs[i]); //считывание в формате float
    scanf("%lf", &mfl[i]); //считывание в формате double

    showShort(ms);
    showInt(ml);
    showLong(mq);
    showFloat(mfs);
    showDouble(mfl);
}

#include <stdio.h>

```

```

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

void showShort(unsigned short arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%hd ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%hX ", arr[i]);
    printf("\n\n");
}

void showInt(unsigned int arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%X ", arr[i]);
    printf("\n\n");
}

void showLong(unsigned long arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%ld ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%lX ", arr[i]);
    printf("\n\n");
}

void showFloat(float arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%f ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%a ", arr[i]);
    printf("\n\n");
}

void showDouble(double arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%lf ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%la ", arr[i]);
    printf("\n\n");
}

int main() {
    unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
    unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF,
                        0xDEADBEEF};
    unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                        0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                        0xBECACA01FFEDC0CA};
    float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
    double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};

    int i = 2;

```

```

scanf("%hu", &ms[i]); //считывание беззнакового целого 16 бит
scanf("%du", &ml[i]); //считывание беззнакового целого 32 бита
scanf("%lu", &mq[i]); //считывание беззнакового целого 64 бита
scanf("%f", &mfs[i]); //считывание в формате float
scanf("%lf", &mfl[i]); //считывание в формате double

showShort(ms);
showInt(ml);
showLong(mq);
showFloat(mfs);
showDouble(mfl);
}

```

1 2 3 4.4 5.5

Вывод массива в знаковом десятичном виде:

-16162 -16162 1 -16162 -16162

Вывод массива в шестнадцатеричном виде:

C0DE C0DE 1 C0DE C0DE

Вывод массива в знаковом десятичном виде:

-559038737 -559038737 2 -559038737 -559038737

Вывод массива в шестнадцатеричном виде:

DEADBEEF DEADBEEF 2 DEADBEEF DEADBEEF

Вывод массива в знаковом десятичном виде:

-4698721151270141750 -4698721151270141750 3 -4698721151270141750 -4698721151270141750

Вывод массива в шестнадцатеричном виде:

BECACA01FFEDC0CA BECACA01FFEDC0CA 3 BECACA01FFEDC0CA BECACA01FFEDC0CA

Вывод массива в знаковом десятичном виде:

2.718282 2.718282 4.400000 2.718282 2.718282

Вывод массива в шестнадцатеричном виде:

0x1.5bf0a8p+1 0x1.5bf0a8p+1 0x1.19999ap+2 0x1.5bf0a8p+1 0x1.5bf0a8p+1

Вывод массива в знаковом десятичном виде:

2.718282 2.718282 5.500000 2.718282 2.718282

Вывод массива в шестнадцатеричном виде:

0x1.5bf0a8b145769p+1 0x1.5bf0a8b145769p+1 0x1.6p+2 0x1.5bf0a8b145769p+1 0x1.5bf0a8b145769p+1

Итог: значение для каждого массива введены корректно, значения соседних элементов остались прежними.

Задание 4.

Для одного из массивов (по варианту) M введите с клавиатуры новое значение всех пяти элементов при помощи одного вызова функции `libc scanf()`.

Проанализировав возвращённое `scanf()` значение, определите корректность ввода; при необходимости отобразите сообщение о количестве не заданных элементов. Выведите массив на экран снова.

Варинт 3

3	Mq
---	------

```

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

void showLong(unsigned long arr[]) {
    printf("Вывод массива в знаковом десятичном виде: \n");
    for (int i = 0; i < 5; i++) printf("%ld ", arr[i]);
    printf("\n");
    printf("Вывод массива в шестнадцатеричном виде: \n");
    for (int i = 0; i < 5; i++) printf("%lX ", arr[i]);
    printf("\n\n");
}

int main() {
    unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                          0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                          0xBECACA01FFEDC0CA};
    while(scanf("%lu %lu %lu %lu %lu", &mq[0], &mq[1], &mq[2], &mq[3], &mq[4]) != 5) {
        puts("Некорректный ввод, попробуйте снова:");
        while (getchar() != '\n');
    }

    showLong(mq);
}

```

```

5 6 7 1 8
Некорректный ввод, попробуйте снова:
2 2 -4 3 2
Вывод массива в знаковом десятичном виде:
2 2 -4 3 2
Вывод массива в шестнадцатеричном виде:
2 2 FFFFFFFF FFFFFFFF FC 3 2

```

Итог: значения для каждого массива введены корректно.

Задание 5.

Введите с клавиатуры при помощи функций `libc`:

- слово (строку без пробелов) s_1 ;
- строку, возможно, содержащую пробелы s_2 ;
- слово s_3 таким образом, чтобы принимающий его буфер

гарантированно не переполнился (если буфер длины k — вводить не более $k - 1$ символов, при необходимости добавляя завершающий нулевой символ).

Выведите на экран при помощи функций libc строки «*** s1 ***», «*** s2 ***», «*** s3 ***» (между звёздочками должна быть введённые строки, а не литералы s1-s3).

```
#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

int main() {
    char str1[30];
    char str2[30];
    char str3[30];
    char x[1];

    scanf("%[^ ]s", str1);
    scanf("%[ ]", x); // избавляемся от пробела
    scanf("%[^|]s", str2); // ввод строки, содержащей пробелы до '|'
    scanf("%[|]", x); // избавляемся от '|'
    scanf("%5c", str3); // ввод последовательности произвольных символов,
                        // ограниченной максимальной шириной поля ввода
    printf("***%s***\n", str1);
    printf("***%s***\n", str2);
    printf("***%s***\n", str3);
}

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

int main() {
    char str1[30];
    char str2[30];
    char str3[30];
    char x[1];

    scanf("%[^ ]s",str1);
    scanf("%[ ]",x); //избавляемся от пробела
    scanf("%[^|]s",str2); //ввод строки, содержащей пробелы до '|'
    scanf("%[|]",x); //избавляемся от '|'
    scanf("%5c",str3); //ввод последовательности произвольных символов, ограниченной максимальной
    шириной поля ввода
    printf("***%s***\n", str1);
    printf("***%s***\n", str2);
    printf("***%s***\n", str3);
}
```

Задание 6.

Выведите на экран при помощи функций libc массивы $M_s \dots M_{fl}$ как таблицу из пяти строк и N столбцов (младшая цифра под младшей цифрой).

```

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

void showShort(unsigned short arr[]) {
    for (int i = 0; i < 5; i++) printf("%25hd ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25hX ", arr[i]);
    printf("\n\n");
}

void showInt(unsigned int arr[]) {
    for (int i = 0; i < 5; i++) printf("%25d ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25X ", arr[i]);
    printf("\n\n");
}

void showLong(unsigned long arr[]) {
    for (int i = 0; i < 5; i++) printf("%25ld ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25lX ", arr[i]);
    printf("\n\n");
}

void showFloat(float arr[]) {
    for (int i = 0; i < 5; i++) printf("%25.8f ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25.5a ", arr[i]);
    printf("\n\n");
}

void showDouble(double arr[]) {
    for (int i = 0; i < 5; i++) printf("%25.8lf ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25.5la ", arr[i]);
    printf("\n\n");
}

int main() {
    unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
    unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF,
                        0xDEADBEEF};
    unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                        0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                        0xBECACA01FFEDC0CA};
    float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
    double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};

    showShort(ms);
    showInt(ml);
    showLong(mq);
    showFloat(mfs);
    showDouble(mfl);
}

#include <stdio.h>

#include <cmath>
#include <iomanip>
#include <iostream>
using namespace std;

void showShort(unsigned short arr[]) {
    for (int i = 0; i < 5; i++) printf("%25hd ", arr[i]);
    printf("\n");
    for (int i = 0; i < 5; i++) printf("%25hX ", arr[i]);

```

```

        printf("\n\n");
    }
    void showInt(unsigned int arr[]) {
        for (int i = 0; i < 5; i++) printf("%25d ", arr[i]);
        printf("\n");
        for (int i = 0; i < 5; i++) printf("%25X ", arr[i]);
        printf("\n\n");
    }
    void showLong(unsigned long arr[]) {
        for (int i = 0; i < 5; i++) printf("%25ld ", arr[i]);
        printf("\n");
        for (int i = 0; i < 5; i++) printf("%25lX ", arr[i]);
        printf("\n\n");
    }
    void showFloat(float arr[]) {
        for (int i = 0; i < 5; i++) printf("%25.8f ", arr[i]);
        printf("\n");
        for (int i = 0; i < 5; i++) printf("%25.5a ", arr[i]);
        printf("\n\n");
    }
    void showDouble(double arr[]) {
        for (int i = 0; i < 5; i++) printf("%25.8lf ", arr[i]);
        printf("\n");
        for (int i = 0; i < 5; i++) printf("%25.5la ", arr[i]);
        printf("\n\n");
    }
    int main() {
        unsigned short ms[] = {0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE, 0xC0DE};
        unsigned int ml[] = {0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF, 0xDEADBEEF,
                             0xDEADBEEF};
        unsigned long mq[] = {0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                              0xBECACA01FFEDC0CA, 0xBECACA01FFEDC0CA,
                              0xBECACA01FFEDC0CA};
        float mfs[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};
        double mfl[] = {exp(1), exp(1), exp(1), exp(1), exp(1)};

        showShort(ms);
        showInt(ml);
        showLong(mq);
        showFloat(mfs);
        showDouble(mfl);
    }

```

-16162 C0DE	-16162 C0DE	-16162 C0DE	-16162 C0DE	-16162 C0DE
-559038737 DEADBEEF	-559038737 DEADBEEF	-559038737 DEADBEEF	-559038737 DEADBEEF	-559038737 DEADBEEF
-4698721151270141750 BECACA01FFEDC0CA	-4698721151270141750 BECACA01FFEDC0CA	-4698721151270141750 BECACA01FFEDC0CA	-4698721151270141750 BECACA01FFEDC0CA	-4698721151270141750 BECACA01FFEDC0CA
2.71828175 0x1.5bf0ap+1	2.71828175 0x1.5bf0ap+1	2.71828175 0x1.5bf0ap+1	2.71828175 0x1.5bf0ap+1	2.71828175 0x1.5bf0ap+1
2.71828183 0x1.5bf0bp+1	2.71828183 0x1.5bf0bp+1	2.71828183 0x1.5bf0bp+1	2.71828183 0x1.5bf0bp+1	2.71828183 0x1.5bf0bp+1

Вопросы

1. Какие функции libc используются для форматированного ввода/вывода?

scanf()/printf()

2. Как задаётся формат ввода/вывода для *scanf()/printf()*?

Ввод

Форматная строка функции *scanf()* может содержать:

- пробельные символы, при этом любая комбинация любых пробельных символов (собственно пробела, табуляции, перевода строки и т. п.) в форматной строке соответствует любой, в том числе другой, вводимой комбинации любых пробельных символов либо их полному отсутствию в указанном месте;
- форматные поля, каждое из которых соответствует последовательности символов, которая может быть преобразована в некоторое значение (в частности, слово, символ, число), а значение — записано по одному из аргументов-адресов;
- иные символы, которые должны присутствовать во вводимых данных «как есть», в противном случае чтение прервётся и *scanf()* завершит работу.

Форматное поле *scanf()* начинается, как и для *printf()*, знаком процента % и заканчивается обозначением формата (таблица 7.4).

Вывод

Форматное поле начинается с символа % и заканчивается буквой, обозначающей обобщённый вид формата, в соответствии с которым выводится значение (таблица 7.2). Символ процента можно вывести с помощью поля %, которому не должно соответствовать никакого параметра функции *printf()*.

Между символом процента и форматом вывода могут присутствовать (но не обязательно) символы, уточняющие формат. Они располагаются в следующем порядке:

- флаги (таблица 7.3; флагов может быть несколько, один или ни одного);
- минимальная ширина поля вывода — число, звёздочка * (значение задаётся параметром) или другое описанное в документации значение;
- точность вывода (для целых форматов — общее количество выводимых цифр, для вещественных — после запятой); точность описывается аналогично мини-

3. Как задаётся размер вводимых/выводимых данных для *scanf()/printf()*?

-//-