ФГОБУ ВПО "СибГУТИ"
Кафедра вычислительных систем

Дисциплины
"ЯЗЫКИ ПРОГРАММИРОВАНИЯ"
"ПРОГРАММИРОВАНИЕ"

Практическое занятие №6

Ввод информации

Преподаватель:

Доцент Кафедры ВС, к.т.н.

Поляков Артем Юрьевич

Шаблон документа

Уважаемый Иван Иванович!

Поздравляем Вас с наступающим Новым Годом, желаем Вам исполнения всех планов и счастья, а Вашей компании ООО «Шаг в себя» - процветания!

Мы, в свою очередь, рады сообщить, что наша компания теперь имеет возможность предоставлять своим клиентам беспроцентные кредиты до 50 000$ для покупки нашего оборудования. Для получения кредита необходим лишь паспорт.

С уважением,

генеральный директор,
Адвертайкин Г.Ц.

Общий для всех писем текст

В котором изменяются только персональные данные

	A	B	C	D	E	F
1	Код	Фамилия	Имя	Отчество	Пол	Компания
2	236	Алхимов	Александр	Сергеевич	ый	ПБЮЛ "Алхимов А.А."
3	93	Белявский	Сергей	Петрович	ый	"Белявский и Ко."
4	87	Петрова	Настасья	Ивановна	ая	ТОО "ИнноТрейд"
5	678	Васильева	Евгения	Петровна	ая	ООО "Хозтовары"
6	789	Иранков	Николай	Иванович	ый	ООО Магазин "Все для дома"
7	117	Песцов	Петр	Александрович	ый	НПО "Монолит"
8	30	Никитаева	Василиса	Алексеевна	ая	ПБЮЛ "Никитаева"
9	76	Дружинин	Евгений	Иванович	ый	ПБЮЛ "Дружинин"

http://www.interface.ru/home.asp?artId=26132

Вывод данных

printf (**PRINT F**ormatted) – функция, обеспечивающая вывод информации на экран компьютера (стандартный вывод).

Функции **printf** необходимо передать следующие **аргументы**:

1. Строку, содержащую шаблон текста, выводимого на экран. Шаблон содержит **константные элементы**, которые выводятся без изменения, а также **спецификаторы**, которые заменяются значениями переменных, которые следуют за строкой.

2. Список переменных (разделитель – запятая), значения которых будут использоваться для замены спецификаторов форматной строки. Переменные сопоставляются спецификаторам в том порядке, в котором они указаны в списке.

```
int x;
float y;
printf("Check: x = %d, y = %f", x, y);
```

`%f` – спецификатор типа

Форматные спецификаторы

```
%d – целый знаковый параметр (int / signed int)
%u – целый беззнаковый параметр (unsigned int)
%f –  вещественный параметр одинарной точности (float)
```

Для параметров других типов используются флаги, которые добавляются к одному из трех типов, указанных выше:

```
%hhd – тип char
%hd – тип short
%ld – тип long
%hhu – тип unsigned char
%hu – тип unsigned short
%lu – тип unsigned long
%lf – тип double
%Lf – тип long double
```

Справочное руководство (printf)

```
$ man 3 printf
```

Для работы с описанными функциями нужно подключить файл **stdio.h**

```
#include <stdio.h>

int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int sprintf(char *str, const char *format, ...);
int snprintf(char *str, size_t size, const char *fmt, ...);
```

Функции вывода в стандартной библиотеке языка Си имеют схожие прототипы (формат вызова):

printf – вывод на экран
fprintf – вывод в файл
sprintf – вывод в строку
snprintf – "безопасный" вывод в строку с ограничением ее размера.

В ОС Unix/Linux экран – это файл, поэтому
printf (…)= fprintf(stdout, …)

C06.1 Вывод данных

c06_1.c

```
#include <stdio.h>
// Output demo program
int main()
{
    int i = 10;
    char c = 19;
    unsigned int ui = 250;
    float f = 120.5;
    double d = 1240.5461;

    // Output variables:
    printf("Integers: i = %d, c = %hhi, ui = %u\n",
           i, c, ui);
    printf("Reals: f = %f, d = %lf\n", f, d);

    return 0;
}
```

C06.2 Переход на новую строку

Символ перехода на новую строку в языке Си обозначается как `'\n'` и может быть использован в любом месте форматной строки:

c06_2.c

```
#include <stdio.h>
// Output demo program
int main()
{
    int i = 10;
    char c = 19;
    float f = 120.5;

    // Output variables:
    printf("Integers: i = %d, c = %hhi\nReals: f = %f\n",
              i, c, f);
    return 0;
}
```

C06.3 Ограничение точности для вещественных типов данных

Вещественные спецификаторы позволяют указать количество знаков после запятой, которое следует выводить:

c06_3.c

```
#include <stdio.h>
// Output demo program
int main()
{
    float f = 120.5564;
    double d = 1.23456;

    // Output variables:
    printf("Original: f = %f, d = %lf\n"
           "Rounded: f = %.2f, d = %.4lf\n", f, d, f, d);
    return 0;
}
```

```
$./c06_3
Original: f = 120.556396, d = 1.234560
Rounded: f = 120.56, d = 1.2346
```

C06.3 Ограничение точности для вещественных типов данных (2)

Вещественные спецификаторы позволяют указать количество знаков после запятой, которое следует выводить:

c06_3.c

```
#include <stdio.h>
// Output demo program
int main()
{
    float f = 120.5564;
    double d = 1.23456;

    // Output variables:
    printf("Original: f = %f, d = %lf\n"
           "Rounded: f = %.2f, d = %.4lf\n", f, d, f, d);
    return 0;
}
```

Почему не 120.5564?

```
$./ c06_3
Original: f = 120.556396, d = 1.234560
Rounded: f = 120.56, d = 1.2346
```

C06.4 Преобразования между системами счисления

Функция **printf** имеет спецификаторы **%x** (he**x**adecimal) и **%o** (**o**ctal), которые позволяют выводить числа в шестнадцатеричной и восьмеричной системах счисления:

c06_4-cvt.c

```
#include <stdio.h>
// Convert demo program
int main()
{
    int i = 58, j = 126, k = -15;
    // Output variables:
    printf("i: %d %u %x %o\n", i, i, i, i);
    printf("j: %d %u %x %o\n", j, j, j, j);
    printf("k: %d %u %x %o\n", k, k, k, k);
    return 0;
}
```

```
$ ./c06_4-cvt
i: 58 58 3a 72
j: 126 126 7e 176
k: -15 4294967281 fffffff1 37777777761
```

C06.4 Преобразования между системами счисления (2)

Функция **printf** имеет спецификаторы **%x** (he**x**adecimal) и **%o** (**o**ctal), которые позволяют выводить числа в шестнадцатеричной и восьмеричной системах счисления:

c06_4-cvt.c

```
#include <stdio.h>
// Convert demo program
int main()
{
    int i = 58, j = 126, k = -15;
    // Output variables:
    printf("i: %d %u %x %o\n", i, i, i, i);
    printf("j: %d %u %x %o\n", j, j, j, j);
    printf("k: %d %u %x %o\n", k, k, k, k);
    return 0;
}
```

```
$ ./c06_4-cvt
i: 58 58 3a 72
j: 126 126 7e 176
k: -15 4294967281 fffffff1 37777777761
```

Откуда эти числа?

Табуляция

Горизонтальная табуляция (HT, TAB) — управляющий символ таблицы ASCII с кодом 0916, используется для выравнивания текста в строках. Встретив этот символ, терминал перемещает каретку (или курсор) вправо на ближайшую позицию табуляции. Традиционно эти позиции располагаются каждые 8 знакомест, в колонках 1, 9, 17, 25… Вводится при помощи клавиши Tab ⇆ , во многих языках программирования обозначается как `'\t'` (В частности, в языке Си).

Для текста слева (символ табуляции обозначен стрелкой) будет получен результат, приведенный справа:

```
one→two→three→four
1→2→3→4
5→6→7→8
9→10→11→12
```

```
one     two     three   four
1       2       3       4
5       6       7       8
9       10      11      12
```

C06.5 Применение табуляции

Напишите программу **c06_5.c**, распечатывающую приведенный ниже текст.

```
one     two     three   four
1       2       3       4
5       6       7       8
9       10      11      12
```