

Last updated: 2020-08-04

vadimov@i.ua

Занятие #1.

Задача 1.1. В следующей программе используется математическая (math) библиотека.

Правильно откомпилируйте и запустите ее. Примечание: здесь используется функция erf, чтобы вычислить площадь популярной у статистиков области (95-процентный доверительный интервал для критерия проверки гипотезы о нормальном распределении).

```
$ cat erf.c
```

```
#include <math.h> //erf, sqrt
#include <stdio.h> //printf
int main () {
    printf("The integral of a Normal (0,1) distribution "
           "between -1.96 and 1.96 is: %g\n", erf(1.96 * sqrt(1/2.)));
}
$ gcc erf.c ...?
```

```
$ ./erf.exe
```

```
The integral of a Normal (0,1) distribution between -1.96 and 1.96 is: 0.950004
```

Задача 1.2. Посмотрите в /lib и /usr/lib какие библиотеки доступны.

- Используйте команду man для получения подробной информации о функциях библиотеки.
- Изучите библиотеки, чтобы увидеть, что в них содержится, запустив команду ar t libfile.

Задача 1.3. Предположим, у вас есть программа на C, основная функция которой находится в main.c и содержит другие функции в файлах input.c и output.c:

- Какие команды вы бы использовали в своей системе для компиляции и компоновки этой программы?
- Как бы вы изменили вышеупомянутые команды, чтобы связать библиотеку с именем process1, хранящуюся в стандартном системном каталоге библиотеки?
- Как бы вы изменили приведенные выше команды, чтобы связать библиотеку с именем process2, хранящуюся в вашем домашнем каталоге?
- Некоторые файлы заголовков необходимо прочитать, и они находятся в подкаталоге заголовка вашего домашнего каталога, а также в текущем рабочем каталоге. Как бы вы изменили команды компилятора, чтобы учесть это?

Задача 1.4. Предположим, у вас есть программа на C, состоящая из нескольких отдельных файлов, и они используют заголовочные файлы, как показано ниже:

main.c - stdio.h, process1.h

input.c - stdio.h, list.h

output.c - stdio.h
process1.c - stdio.h, process1.h
process2.c - stdio.h, list.h

- Какие файлы нужно перекомпилировать после внесения изменений в process1.c?
- Какие файлы нужно перекомпилировать после внесения изменений в process1.h?
- Какие файлы нужно перекомпилировать после внесения изменений в list.h?

Задача 1.5. Напишите программу на языке ANSI C. В качестве входных данных введите целое число секунд, а в качестве выходных данных выведите эквивалентное время в часах, минутах и секундах. Рекомендуемый формат вывода:
7322 секунды эквивалентны 2 часам 2 минутам 2 секундам.

Задача 1.6. Напишите программу на языке ANSI C, которая будет читать из stdin произвольное целое число - основание системы счисления, затем положительное целое число, записанное в этой системе счисления, и выводить его значение в stdout. Читайте второе целое число за раз; пропустите любые начальные недопустимые символы, затем считывайте допустимые символы, пока не встретите недопустимый. Пример:

Input	Output
10 1234	1234
8 77	63 (the value of 77 in base 8, octal)
2 1111	15 (the value of 1111 in base 2, binary)

Основание системы счисления меньше или равно 10.

Задача 1.7. Напишите программу на языке ANSI C для заполнения массива случайными числами с плавающей точкой. Напишите функцию для поиска заданного числа в этом массиве.

Задача 1.8. Напишите функцию "replace", которая получает указатель на строку в качестве параметра. Функция заменяет все пробелы в этой строке знаками минус, и возвращает количество пробелов, которые она заменила. Пример:

```
char *cat = "The cat sat";  
n = replace(cat);
```

должна установить

```
cat to "The-cat-sat"
```

и вернуть n равное 2.

Задача 1.9. Напишите программу с использованием перечислимых типов, которая, при задании сегодняшней даты будет распечатывать завтрашнюю дату.

Задача 1.10. Напишите простую программу базы данных на языке ANSI C, в которой будут

хранятся такие данные, как возраст, дата рождения, адрес и т. п. Создавать свою RDBMS не требуется. :)

Задача 1.11. Напишите программу на языке ANSI C для заполнения массива произвольного типа случайными значениями с использованием указателей. Напишите функцию поиска заданного значения в этом массиве.

Задача 1.12. Напишите программу, которая читает число, которое задает размер целочисленного массива, создает массив, соответствующего размера, а затем заполняет этим значением массив. Подсказка: не забудьте освободить память.

Задача 1.13. Напишите программу `last`, которая печатает последние `n` строк своего текстового ввода. По умолчанию `n` должно быть 5, но ваша программа должна разрешать необязательный аргумент так, чтобы `last -n` выводила последние `n` строк, где `n` - любое целое число. Ваша программа должна наилучшим образом использовать память! (Ввод текста может быть путем чтения файла, указанного в команде, или чтения файла из стандартного ввода)

Задача 1.14. Напишите программу, которая сортирует список целых чисел в порядке возрастания. Однако, если в командной строке присутствует флаг `-r`, ваша программа должна отсортировать список в порядке убывания. (Вы можете использовать любую процедуру сортировки, какую пожелаете).

Задача 1.15. Напишите программу, которая читает следующую структуру и сортирует данные по `keyword`, используя `qsort`:

```
typedef struct {
    char keyword[10];
    int other_data;
} Record;
```

Задача 1.16. Сортировка методом вставки выполняется путем последовательного добавления значений в массив. Первое значение просто хранится в начале массива. Каждое последующее значение добавляется путем нахождения его упорядоченной позиции в массиве, перемещения данных по мере необходимости для размещения значения и вставки значения в эту позицию. Напишите функцию с именем `insort`, которая выполняет эту задачу и ведет себя так же, как `qsort`, то есть она может сортировать массив по значению ключа любого типа и имеет аналогичный прототип.

Задача 1.17. Напишите функцию, которая печатает 8-битное (без знака) число в двоичном формате.

Задача 1.18. Напишите функцию, которая инвертирует биты беззнакового символьного `x` и сохраняет ответ в `y`. Распечатать результат в двоичной форме (см. предыдущую задачу). Ввод может быть и в десятичной форме. Ваш вывод должен быть таким:

```
x = 10101010 (binary)
x inverted = 01010101 (binary)
```

Задача 1.19. Создайте макроопределение `swap(t, x, y)`, которое будет осуществлять обмен значений двух аргументов `x` и `y` заданного типа `t`.

Задача 1.20. Напишите программу, которая читает заданное число из командной строки и генерирует случайное число с плавающей запятой в диапазоне от 0 до заданного числа.

Задача 1.21. Установите в своей ОС компилятор `clang` (<https://clang.llvm.org/>). Откомпилируйте программу `hello.c`. Сравните код, сгенерированный компилятором `clang`, с кодом, который был сгенерирован компилятором `gcc`. В чем отличия?

```
$ clang --version
clang version 8.0.1 (tags/RELEASE_801/final)
Target: x86_64-unknown-windows-cygnus
Thread model: posix
InstalledDir: /usr/bin

$ make hello_dbg
clang -g -ggdb -gdwarf-4 -O0 -Wall -Wextra hello.c -o hello_dbg
objdump --source ./hello_dbg > hello_dbg_objdump.txt

$ cat hello_dbg_objdump.txt | grep -B 3 -A 2 "a = 5"
. . .
```