#include <stdio.h> \*\*включить стандартную библиотеку ввода-вывода (на языке препроцесора си) std input output (.h заголовочных header)

int main (int argc, char\* argv[]) \*\*функция одна на файл, запускает инетрпритатор командной строки

{

printf(“Hello %s!n”, “John”); \*\*

return 0; \*\* код ошибки 0

}

\n перевод каретки

\t табуляция (отступ)

\\ = \

спецификатор формата

%s – вставка строки (не надо ставить & в остальных обязательно)

%d – для чисел

компилятор преобразует исходный текст в исполняемый файл

компиляция

#include <stdio.h>

int main()

{

printf(“I bot, You name?\n”);

char name[20];

scanf(“%s”, name); \*\*%s – строка должна быть массивом символов. (не надо ставить вэто

printf(“Hello, %s, How old are you?\n”, name);

int age;

scanf(“%d”, &age); \*\*%d -

printf(“you are younger!”

“I though you are %d!\n”, age-3);

return 0;

}

#include <unistd.h> \*\*включить бибилиотека вывода символов. (препроцесор)

void ft\_putchar(char c) \*\* putchar, не возвращает значения

\*\* преобразует переменную (с) с типом данных char, с вывовод на экран (таблицу ASCIII), нигде не сохраняя

{

write(1, &c, 1); \*\*char c передает в блок кода и возвращает (если мы подставим 97, то получим символ а)

} \*\*&c – ссылка на переменную, с – сокращение char

write (<дескриптор.потока>, <&указатель на буфер>, <число.записыв.байтов>):

0 stdin – стандартное устройство ввода (клавиатура);

1 stdout – стандартное устройство вывода (экран);

2 stderr – стандартное устройство вывода сообщения об ошибках (также экран).

const void \*buf <указатель на буфер> - адрес, куда мы будем записывать наши данные (байты) и возращать их.

Значение из ft\_putchar(char c(отсюда)) передав его в write(1, &c(cюда), 1).

Синтаксис, когда мы делаем ссылку на адрес с → &c.

size\_t count <число записываемых байтов> - количество пересылаемых данных (1 byte - 1 char).

Эта функция записывает один символ в стандартный поток вывода терминала.

На выходе имеем функцию write – с параметрами вывода паременной (с) на экран по 1 символу.

1. Создайте функцию, которая будет выводить алфовит с маленькой буквы, в одну строку, в порядке возрастания, начиная с буквы ‘a’.

Create a function that displays the alphabet in lowercase, on a single line, by ascending order, starting from the letter ’a’.

void ft\_print\_alphabet(void) \*\*Объявляем функцию void - прототип, мы ее не будем потом исользовать, мы только вычисляем в ней.

{

char a; \*\* Объявляем целочисленную переменную a

\*\* Присваиваем a=97 (a - просто ярлык, присваиваю ему номер 97 из ASCII таблице, который обозначает символ (a)

\*\* ( переменная a ничего общего с ASCII таблицей не имеет, я могу и b поставить)

a = 97;

while (a <= 122) \*\* Цикл: пока (a / 97) меньше или равна (122) выполняем тело цикла

{

ft\_putchar(a); \*\* Выполняем цикл с ft\_putchar, поместив туда (a) = (97) будет выведена на экран в типе char 1 раз)

a++; \*\* Выводить (a) как? a(97) = (97) + 1. Тобишь присваиваем a + 1. Этот метод опис. в видео.

}

}

main.c

int main() \*\* Внутри функции main находится наша программа (void – нет параметров)

{

ft\_print\_alphabet(); \*\*референс

return (0); \*\* в main это команда возвращает значение вызыающей фнкцию и заканчивает фнкцию main.

}