ВОПРОС 1.

Как соотносится содержимое узла Sourses с содержимым папки S:\ЛР 3\FarToCels\?

Содержимое узла Sourses, то есть файлы проекта FarToCels с расширением .c, находится в Workspace\FarToCels\Sources и это по сути тоже самое, что и файлы директории S:\ЛР 3\FarToCels\ с расширением .c, только помимо этого в S:\ЛР 3\FarToCels\ лежат папки Дебага и Релиза(каждая в папках bin и obj)

Иными словами Sources в среде CodeBlocks это собственное подмножество файлов проекта действительной директории на самом компьютере(в данном случае это FarToCels). Причём его же можно посмотреть во вкладке Files интегрированной среды разработки CodeBlocks.

ВОПРОС 2. Что произошло при нажатии на кнопку Build?

Проект собрался и скомпилировался.

ВОПРОС 3. Что за файлы появились в папках S:\ЛР3\FarToCels\bin\Debug\ и S:\ЛР3\FarToCels\obg\Debug\

В папке S:\ЛР3\FarToCels\bin\Debug\ появился скомпилированный проект FarToCels;

В папке S:\ЛР3\FarToCels\obg\Debug\ появились файлы проекта main.o, Var1.o, Var2.o, Var3.o, Var4.o

Задание 4.1.

Превратите строку 16 текста функции main() в комментарий.

Перекомпилируйте программу (кнопка Build). Попытайтесь объяснить сообщение компилятора, появившееся в нижней части экрана.

Объяснение поместить в отчёт.

Уберите признак комментария из строки 16.

При повторном build'е проекта произошла ошибка expected identifier or '(' before '{' token {

На стадии сборки линкер не смог найти функцию main(), ведь она закомментирована

main должен быть в каждом Си проекте

Задание 4.2.

Запустите проект на исполнение (достаточно нажать на кнопку Build and run

или на F9). Сохраните скриншот консоли до первого нажатия знаменитой кнопки

AnyKey.

Поpressуйте AnyKey до завершения программы. ( Если лень, то нажмите на

кнопку Abort, расположенную там же, где Build. Получите кучу лишних проблем. )

Удалите круглые скобки в строке 23 и нажмите F9.

Сравните появившееся окно с сохранённым скриншотом и попытайтесь объяс-

нить различия. Объяснение поместить в отчёт.

Восстановите удалённые круглые скобки (alt+backspace).

Если объяснение “не выстраивается”, проделайте то же самое для строки 26.

За именем функции обязательно должна следовать пара круглых скобок. В

скобках размещается список параметров функции, если она их имеет. Если парамет-

ров нет, то в скобках пусто. Но скобки есть.

В таком случае, при несоблюдении синтаксиса, мы получаем ошибку компилятора, когда убираем скобки у ФУНКЦИИ getch().

Задание 4.3.

Удалите круглые скобки после имени var1 в строке 16 и нажмите F9. Помести-

те в отчёт объяснение сообщения компилятора в нижней части окна CodeBlocks.

Восстановите удалённые скобки .

Объяснение то же самое, что и при ситуации с getch(), так как идентификатор var1 обозначает функцию var1(), и при удалении скобок функции возникает очевидная ошибка.

Задание 4.4.

Внимательно изучите текст функции main. Напишите в тексте комментарии к

строкам 16, 19, 22 – 26. Комментарии должны пояснять смысл инструкций, как Вы

его понимаете. Если не понимаете, не стесняйтесь в этом признаться. Это (пока) не

страшно .

Скопируйте откомментированный текст в отчёт через буфер обмена Windows.

int main()

{

int var1(), var2();

int var3(), var4();

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Перевод показаний термометра из шкалы Фаренгейта в шкалу Цельсия\n\n");

printf("\tPress anykey\n"); /\*Выводит в консоль сообщение пользователю о просьбе "Press any key"(так сказать предупреждение о дальнейших действиях в программе). В строке использовано форматирование: \t - горизонтальная табуляция; \n - переход каретки на следующую строку.\*/

/\*Чтение одного символа с клавиатуры, этот символ на экран не выводится. Используется чтобы не дать консоли преждевременно закрыться. Эта функция ожидает пока пользователь введёт символ и лишь затем закрывает консоль\*/

getch();

printf("\n\n\tVar 1\n"); /\*Сообщение пользователю о предоставляемых данных функции var1\*/

var1(); //Вызов функции var1()

printf("\n\tPress anykey\n"); /\*Аналогично printf("\tPress anykey\n"); с изменением форматирования текста\*/

getch();

printf("\n\n\tVar 2\n"); /\*Сообщение о предоставляемых данных функции var2()\*/

var2(); //Вызов функции var2()

printf("\n\tPress anykey\n"); //Аналогично printf("\n\tPress anykey\n");

getch();

printf("\n\n\tVar 3\n"); /\*Сообщение о предоставляемых данных функции var3()\*/

var3(); //Вызов функции var3()

printf("\n\tPress anykey\n"); //Аналогично printf("\n\tPress anykey\n");

getch();

printf("\n\n\tVar 4\n"); /\*Сообщение о предоставляемых данных функции var4()\*/

var4(); //Вызов функции var4()

return 0; /\*Возврат целочисленного значения 0 из функции main обратно в операционную систему обозначая успешное завершение программы\*/

}

Задание 4.5.

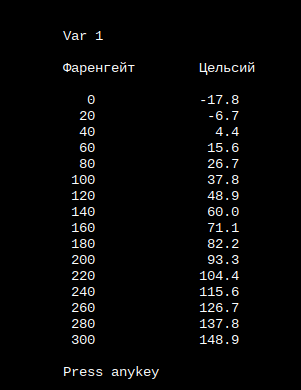
Откройте тексты функций Var1.c и Var2.c, как открывали текст main.c. Изучи-

те тексты и откомментируйте каждую строку. Излагайте СВОЁ понимание инструк-

ций. Если не понимаете, то так и пишите: “Не понимаю”.

Сравните тексты. Проанализируйте различия и дайте комментарии.

Тексты с комментариями и результаты анализа поместите в отчёт.

int var1()

{

int lower,//Объявление переменных целочисленного типа

upper,

step;

float fahr,//Объявление переменных типа с плавающей точкой

celsius,

k,

a;

//Определение переменных

lower=0;

upper=300;

step=20;

k=5.0/9.0;

a=32.0;

//Выводим в консоль сообщение пользователю

printf("\n\tФаренгейт\t Цельсий\n\n");

// Присваиваем переменной fahr значение переменной lower

fahr=lower;

/*\*Запускаем цикл с предусловием таким, что цикл будет перебирать все градусы фаренгейт с шагом step до максимальной границы upper\*/*

while (fahr<=upper)

{

//Получаем значение в градусах Цельсия

celsius=k\*(fahr-a);

/*\*Выводим значение в Фаренгейтах и соответственное ему в Цельсиях\**/

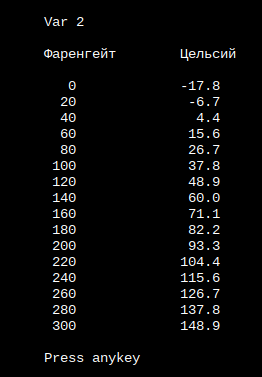
printf("\t%4.0f\t\t%6.1f\n", fahr, celsius);

//Переходим к следующему значению

fahr=fahr+step;

}

return 0;//Возвращаем в функцию main значение 0

}

int var2()

{

//Инициализация переменных целочисленного типа

int lower=0,

upper=300,

step=20;

//Инициализация переменных типа с плавающей точкой

float fahr=lower,

k=5.0/9.0,

a=32.0,

celsius;

//Выводим в консоль сообщение пользователю

printf("\n\tФаренгейт\t Цельсий\n\n");

/*\*Запускаем цикл с предусловием таким, что цикл будет перебирать все градусы фаренгейт с шагом step до максимальной границы upper\*/*

while (fahr<=upper)

{

//Получаем значение в градусах Цельсия

celsius=k\*(fahr-a);

/*\*Выводим значение в Фаренгейтах и соответственное ему в Цельсиях\**/

printf("\t%4.0f\t\t%6.1f\n", fahr, celsius);

//Переходим к следующему значению

fahr=fahr+step;

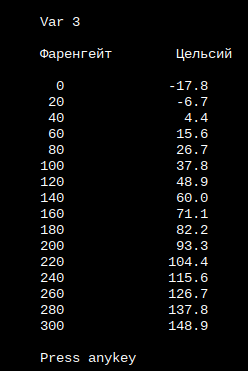
}

return 0;//Возвращаем в функцию main значение 0

}

Отличия: в var1 идёт объявление и определение переменных в отличие от var2, где идёт сразу инициализация переменных.

Задание 4.6.

Проделайте то же самое для текстов Var2.c и Var3.c.

/\*Ввод именованных констант с помощью препроцессорной директивы define с автоматическим определением типа данных\*/

#define K 5.0/9.0

#define A 32.0

#define LOWER 0

#define UPPER 300

#define STEP 20

int var3()

{

//инициализация переменной f

float f=LOWER;

//Выводим в консоль сообщение пользователю

printf("\n\tФаренгейт\t Цельсий\n\n");

/*\*Запускаем цикл с предусловием таким, что цикл будет перебирать все градусы фаренгейт с шагом step до максимальной границы upper\*/*

while (f<=UPPER)

{

/\*Получаем значение в градусах Цельсия и сразу Выводим значение в Фаренгейтах и соответственное ему в Цельсиях\*/

printf("\t%3.0f\t\t%5.1f\n", f, K\*(f-A));

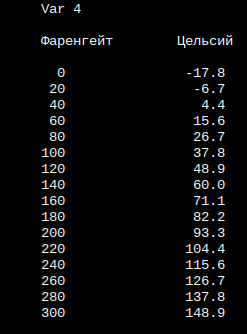
f=f+STEP;

}

return 0;

}

Отличия: в var3 введение констант происходит на препроцессорном уровне, а в var2 на стадии компиляции; в var3 значение градуса в цельсиях происходит при отправке параметра функции printf, то есть сначала вычисляется значение, потом отправляется оно в консоль, в var2 же это выведено в разные стейтменты(операторы).

Задание 4.7.

Проделайте то же самое для текстов Var3.c и Var4.c.

#include <stdio.h>

/\*Ввод именованных констант с помощью препроцессорной директивы define с автоматическим определением типа данных\*/

#define K 5.0/9.0

#define A 32.0

#define LOWER 0.0

#define UPPER 300.0

#define STEP 20.0

int var4()

{

float f;

printf("\n\tФаренгейт\t Цельсий\n\n");

/*\*Запускаем* **парамерический** *цикл перебираюший все градусы фаренгейт с шагом step до максимальной границы upper\*/*

for (f=LOWER; f<=UPPER; f+=STEP)

printf("\t%3.0f\t\t %5.1f\n", f, K\*(f-A));

return 0;

}

Отличия: в var3 используется цикл в предусловием, в var4 параметрический цикл.