**Лабораторная работа № 1.**

**«Программирование. Язык С: общие сведения»**

**Ответы на вопросы.**

**1. Что такое препроцессор, его функции и обращение к нему.**

**Опишите основные библиотеки и их назначение.**

Препроцессора Си - это программа, которая обрабатывает входные данные для компилятора. Препроцессор просматривает исходную программу и выполняет следующие действия: подключает к ней заданные файлы, , осуществляет подстановки, а также управляет условиями компиляции. Для препроцессора предназначены строки программы, начинающиеся с символа #. В одной строке разрешается записывать только одну команду (директиву препроцессора).

<assert.h> Макрос assert()

<ctype.h> Обработка символов

<errno.h> Сообщения об ошибках

<float.h> Значения с плавающей точкой, зависящие от конкретной реализации компилятора

<iso646.h> Макросы, соответствующие различным операторам, например, && и ^. Добавлено в 1995 году (Поправка 1)

<limits.h> Различные ограничения, зависящие от конкретной реализации компилятора

<locale.h> Функция setlocale()

<math.h> Различные определения, используемые библиотекой math

<setjmp.h> Нелокальные переходы

<signal.h> Значения сигналов

<stdarg.h> Списки аргументов переменной длины

<stddef.h> Распространенные константы

<stdiо.h> Ввод-вывод файлов

<stdlib.h> Смешанные объявления

<string.h> Функции обработки строк

<time.h> Функции системного времени и даты

<wctype.h> Функции обработки многобайтовых слов и двубайтовых символов. Добавлено в 1995 году (Поправка 1)

<wchar.h> Функции классификации многобайтовых слов и двубайтовых символов. Добавлено в 1995 году (Поправка 1)

**2. Сколько функций минимально должна содержать программа на**

**С?**

Минимальное количество функций – одна (main).

**3. Что реализует функций main?**

Это самая главная функция в программе, с неё начинается выполнение любой программы на языке Си, т. е. при запуске программы она выполняется первой. Управляет выполнением программы, вызывая другие функции.

**4. Напишите способы задания комментария в программе и укажите**

**на их отличия.**

Комментарий — это последовательность символов, которая начинается с косой черты и звездочки (/\*) и рассматривается компилятором как один пробельный символ, а во всем остальном игнорируется. Комментарии могут включать любое сочетание символов из представимого набора символов, включая символы новой строки, но не включая разделитель конца комментария (\*/). Комментарии могут занимать несколько строк, но не могут быть вложенными.

//Комментарий - однострочный комментарий,

/\* Комментарий программы\*/ - многострочный комментарий.

**5. Оператор присваивания. Дополнительные операции**

**присваивания.**



**6. Оператор вывода. Формат вывода.**

printf("Текст, который вы хотите вывести должен быть заключён между кавычками.");

%x — целое число типа int со знаком в шестнадцатеричной системе счисления;

%o — целое число типа int со знаком в восьмеричной системе счисления;

%hd — целое число типа short со знаком в десятичной системе счисления;

%hu — целое число типа unsigned short;

%d — целое число типа int со знаком в десятичной системе счисления;

%u — целое число типа unsigned int;

%hx — целое число типа short со знаком в шестнадцатеричной системе счисления;

%ld — целое число типа long int со знаком в десятичной системе счисления;

%lu — целое число типа unsigned long int;

%lx — целое число типа long int со знаком в шестнадцатеричной системе счисления;

%f — вещественный формат (числа с плавающей точкой типа float);

%lf — вещественный формат двойной точности (числа с плавающей точкой типа double);

%c — символьный формат;

%s — строковый формат;

%e — вещественный формат в экспоненциальной форме (числа с плавающей точкой типа float в экспоненциальной форме);

**7. Специальные символы, которые используются в выводе и их**

**назначение.**

'\n' - перевод строки;

'\t' - горизонтальная табуляция;

'\v' - вертикальная табуляция;

'\b' - возврат на символ;

'\r' - возврат на начало строки;

'\a' - звуковой сигнал.

**8. Описать алгоритм работы программы:**

**int dir( )/\* dir\*/ -** задание функции

**{** - открытие скобок(начало функции)

**printf("добрый день\n"); -** вывод текста

**printf(" мы рады вас видеть\n"); -** вывод текста

**} –** закрытие скобок **(**конец функции)

**int main ()** – описание функции main, начало программы

**{** - открытие скобки (начало прграммы)

**dir ();** - вызов функции

**printf("надеемся на плодотворную работу \n");** - вывод текста

**printf("и замечательные успехи\n");** - вывод текста

**} -** закрытие скобок **(**конец программы)

**9. Описать все компоненты строки программы:**

**printf("сейчас %d год \n", year);**

**printf** – оператор вывода на печать

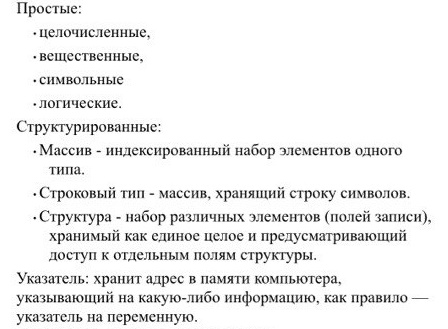
**%d** – спецификатор, на место которого записывается значение переменной, которая является вторым аргументом функции printf

**\n –** осуществляет переход на новую строку.

**year** – второй аргумент функции printf

**10.Какие типы данных существуют. Их обозначение. Запись**

**переменных.**





**11.Назовите ошибку в имени идентификатора 40Hours, Get Data, box-**

**22, cost\_in\_$, int**

**40Hours** – имя идентификатора начинается с цифры.

**Get Data** – имя идентификатора содержатся пробелы.

**box-22** – имя идентификатора содержится знак «-».

**cost\_in\_$** – имя идентификатора не может содержать знак «$».

**int** – имя идентификатора пустое или имя идентификатора не должно совпадать с ключевым словом.

**12.Есть ли разница в идентификаторах Hours и hours (машина**

**воспримет как две переменных или как одну и ту же)?**

Си - язык регистрозависимый. Это означает, что заглавные и строчные буквы, рассматриваются компилятором, как разные сущности.

**13.Выберите из предложенных значений допустимые присваивания**

**для переменных:**

|  |  |
| --- | --- |
| **Переменные:** | **Значения:** |
| **num** | **2856, alpha** |
| **alpha** | **2856** |
| **rate** | **0.36** |
| **ch** | **b, alpha** |

**14.Арифметические операции: сложение, вычитание, умножение,**

**возведение в степень, деление, целочисленное деление, остаток от**

**деления**

**+** - сложение;

**-** - вычитание;

**\*** - умножение;

**double pow (double x, double y); -**

**float powf (float x, float y); -** возведение в степень

**long double powl (long double x, long double y); -**

**(float)x/y;** - целочисленное деление;

**/** - деление;

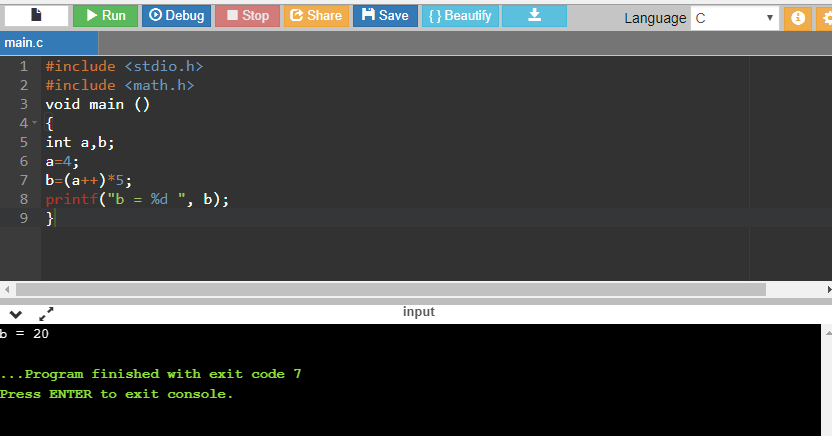
**%** - остаток от целочисленного деления.

**15.Операции инкремента и декремента в С**

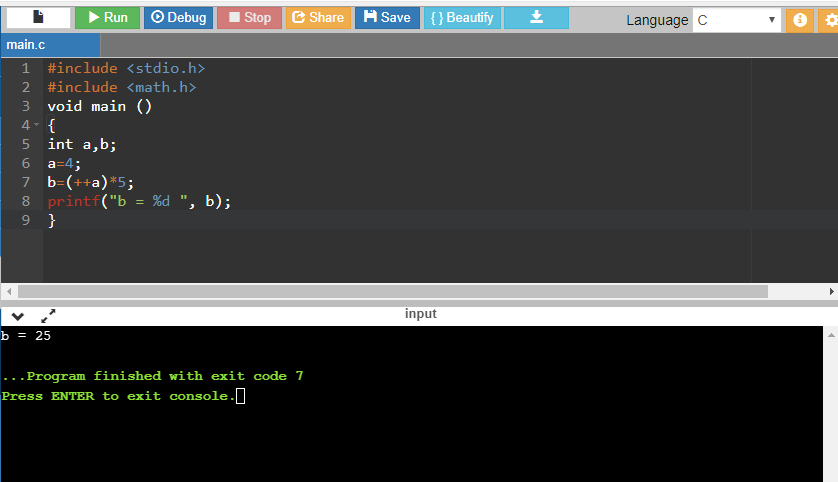
инкрементирование (увеличение на 1) ++ ;

декрементирование (уменьшение на 1) -- ;

**16.Получить результат выражения b = (a++)\*5; (при a=4)**



**17.Получить результат выражения b = (++a)\*5; (при a=4)**



**18.Операции отношения (сравнения): равно, не равно, больше,**

**меньше, больше или равно, меньше или равно**

**== -**равно

**!=**  -не равно

**>**  -больше

**< -**меньше

**>=** -больше или равно

**<= -**меньше или равно

**19.Логические операции: не, и, или**

**! -** НЕ (унарная)- невыполнение операции отношения.

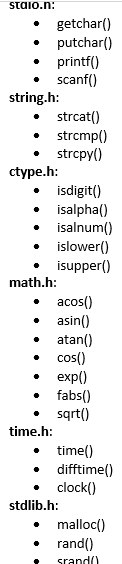
**&& -** И (бинарная)- одновременное выполнение всех операций отношения;

|| - ИЛИ (бинарная) - выполнение хотя бы одной операции отношения;

**20.Оператор ввода с клавиатуры**

scanf()

**21.Библиотечные функции**



**22.Вычислить отношение длинны окружности к площади круга**

#include <stdio.h>

int main()

{

float Dl,S,rezult,r;

printf("Введите радиус окружности \n");

scanf("%f",&r);

Dl=2\*3.14\*r;

printf("Длина окружности= %f \n", Dl);

S=3.14\*r\*r;

printf("Площадь круга= %f \n",S);

rezult=Dl/S;

printf("rezult= %.3f",rezult);

}

