

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Новгородский государственный университет имени Ярослава Мудрого»
Институт электронных и информационных систем
Кафедра радиосистем

УТВЕРЖАЮ
Директор ИЭИС
И.Э. Эминов
« 29 » 01 2019 г.

РАБОЧАЯ ПРОГРАММА

учебной дисциплины
ЯЗЫКИ ПРОГРАММИРОВАНИЯ
Учебный модуль по направлению подготовки
11.03.01 Радиотехника

Направленность (профиль) Радиотехнические средства передачи,
приема и обработки сигналов

11.03.03 Конструирование и технология электронных средств
Направленность (профиль) Проектирование и технология радиоэлектронных
средств

11.03.04 Электроника и нанoeлектроника
Направленность (профиль) Микроэлектроника и твердотельная электроника

СОГЛАСОВАНО

Начальник отдела обеспечения
деятельности ИЭИС

П.В. Лысухо
« 08 » апреля 2019 г.

Заведующий выпускающей
кафедрой КИТР

М.И. Бичурин
« 30 » 01 2019 г.

Заведующий выпускающей
кафедрой ФТТМ

Б.И. Селезнев
« 30 » 01 2019 г.

Разработал

Ассистент кафедры РС

Н.В. Вахлачев
« 29 » 01 2019 г.

Заведующий кафедрой РС

И.Н. Жукова
« 29 » 01 2019 г.

Заведующий кафедрой РС

И.Н. Жукова
Принято на заседании кафедры РС
Протокол № 133 от 20.02.2019 г.

Заведующий кафедрой

И.Н. Жукова
« 30 » 01 2019 г.

1 Цели и задачи освоения учебной дисциплины

Целью освоения учебной дисциплины является формирование компетентности студентов в области информационных технологий, а также в области инженерной и компьютерной графики, способствующей становлению их готовности к решению задач профессиональной деятельности.

Основными задачами являются:

- приобретение студентами знаний в области программирования, освоение возможностей языка *Си* с концентрацией на решении задач инженерного характера;
- эффективно использовать возможности информационных технологий является составной частью профессиональной компетентности современного специалиста в любой сфере деятельности;
- основами компьютерных технологий в области программирования, а также развивать навыки алгоритмизации инженерных задач.

2 Место учебной дисциплины в структуре ОПОП

Учебная дисциплина относится к обязательной части учебного плана основных профессиональных образовательных программ направлений подготовки 11.03.01 Радиотехника и направленности (профилю) Радиотехнические средства передачи, приема и обработки сигналов, 11.03.03 Конструирование и технология электронных средств направленности (профилю) Проектирование и технология радиоэлектронных средств, 11.03.04 Электроника и наноэлектроника направленности (профилю) Микроэлектроника и твердотельная электроника. В качестве входных требований выступают сформированные ранее компетенции обучающихся, приобретенные ими в рамках следующих дисциплин (модулей, практик): «Математика», «Физика». Освоение учебной дисциплины является компетентностным ресурсом для дальнейшего изучения следующих дисциплин (модулей, практик): «Информационные технологии», «Схемотехника аналоговых устройств», «Цифровая обработка сигналов», «Цифровые устройства и микропроцессоры», «Схемотехника», «Проектирование цифровых устройств», «Прикладная информатика», «Схемо- и системотехника электронных средств», «Схемотехника аналоговых электронных устройств», «Информационные технологии проектирования и производства радиоэлектронных средств».

3 Требования к результатам освоения учебной дисциплины

Перечень компетенций, которые формируются в процессе освоения учебной дисциплины:

Общепрофессиональные компетенции:

ОПК-3 Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности;

ОПК-4 Способен применять современные компьютерные технологии для подготовки текстовой и конструкторско-технологической документации с учетом требований нормативной документации

Таблица 1- Результаты освоения учебной дисциплины «Языки программирования»

Код и наименование компетенции	Результаты освоения учебной дисциплины (индикаторы достижения компетенций)		
ОПК-3 Способен применять методы поиска, хранения, обработки, анализа и представления в требуемом формате информации из различных источников и баз данных, соблюдая при этом основные требования информационной безопасности	Знает современные принципы поиска, хранения, обработки, анализа и представления в требуемом формате информации	Умеет решать задачи обработки данных с помощью современных средств автоматизации	Владеет навыками обеспечения информационной безопасности
ОПК-4 Способен применять современные компьютерные технологии для подготовки текстовой и конструкторско-технологической документации с учетом требований нормативной документации	Знает современные интерактивные программные комплексы для выполнения и редактирования текстов, изображений и чертежей	Умеет использовать современные средства автоматизации разработки и выполнения конструкторской документации	Владеет современными программными средствами подготовки конструкторско-технологической документации

4 Структура и содержание учебной дисциплины (модуля)

4.1 Трудоемкость учебной дисциплины (модуля)

4.1.1 Трудоемкость учебной дисциплины для очной формы обучения

Полная трудоемкость учебной дисциплины для очной формы обучения представлена в таблице 2, для заочной формы обучения таблице 3:

Таблица 2 – Трудоемкость учебной дисциплины для очной формы обучения

Виды учебной работы	Всего	Распределение по семестрам
		1
1. Общая трудоемкость учебной дисциплины в зачетных единицах (ЗЕТ)	3	3
2. Контактная работа в академических часах (АЧ):	54	54
а) аудиторная		
Лекции (ЛЕК)	18	18
Практические занятия (ПЗ)	36	36
б) внеаудиторная		
3. Самостоятельная работа в академических часах (АЧ):	54	54
4. в том числе курсовая работа/курсовой проект (при наличии)	-	-
5. Промежуточная аттестация: (зачет; дифференцированный зачет; экзамен)	зачет	зачет

Таблица 3 – Трудоемкость учебной дисциплины для заочной формы обучения

11.03.03 Конструирование и технология электронных средств

Виды учебной работы	Всего	Распределение по семестрам
		2
1. Общая трудоемкость учебной дисциплины в зачетных единицах (ЗЕТ)	3	3
2. Контактная работа в академических часах (АЧ):	12	12
а) аудиторная		
Лекции (ЛЕК)	4	4
Практические занятия (ПЗ)	8	8
б) внеаудиторная		
3. Самостоятельная работа в академических часах (АЧ):	96	96
4. в том числе курсовая работа/курсовой проект (при наличии)	-	-
5. Промежуточная аттестация: (зачет; дифференцированный зачет; экзамен)	зачет	зачет

4.2 Содержание учебной дисциплины (модуля)

- 1 Общее знакомство. Происхождение языка Си.
- 2 Знакомство с указателями, классы памяти и разработка программ
- 3 Использование текстового редактора для подготовки программ.
- 4 Примеры простой программы. Структура простой программы.
- 5 Структурное программирование.
- 6 Библиотеки языка Си.
- 7 Функции и макроопределения, распределение памяти, примеры функций.
- 8 Массивы и указатели, символьные строки и функции над ними. Структуры.
- 9 Циклы и другие управляющие средства, создание и использование функций

4.3. Трудоемкость разделов учебной дисциплины и контактной работы

Трудоемкость освоения отдельных разделов и тем учебной дисциплины «Языки программирования» представлены в таблице 4.

Таблица 4 – Трудоемкость освоения разделов и тем учебной дисциплины «Языки программирования»

№	Наименование разделов учебной дисциплины (модуля), УЭМ, наличие КП/КР	Контактная работа (в АЧ)				Внеауд. СРС (в АЧ)	Формы текущего контроля
		Аудиторная			В т.ч. СРС		
		ЛЕК	ПЗ	ЛР			
1	Общее знакомство. Происхождение языка <i>Си</i>	2	4		1	6	Опрос
2	Знакомство с указателями, классы памяти и разработка программ	2	4		1	6	Опрос, практические задания
3	Использование текстового редактора для подготовки программ.	2	4		1	6	Опрос, практические задания
4	Примеры простой программы. Структура простой программы.	2	4		1	6	Опрос, практические задания
5	Структурное программирование	2	4		1	6	Опрос, практические задания
6	Библиотеки языка <i>Си</i>	2	4		1	6	Опрос, практические задания
7	Функции и макроопределения, распределение памяти, примеры функций	2	4		1	6	Опрос, практические задания
8	Массивы и указатели, символьные строки и функции над ними. Структуры	2	4		1	6	Опрос, практические задания
9	Циклы и другие управляющие средства, создание и использование функций	2	4		1	6	Опрос, практические задания
	Промежуточная аттестация						зачет
	ИТОГО	18	36	0	9	54	

4.4 Лабораторные работы и курсовые работы/курсовые проекты

4.4.1 Перечень тем лабораторных работ:

Лабораторные работы не предусмотрены учебным планом.

4.4.2 Примерные темы курсовых работ/курсовых проектов:

Курсовые работы/курсовые проекты не предусмотрены учебным планом.

5 Методические рекомендации по организации освоения учебной дисциплины освоения учебной дисциплины

Таблица 5 – Методические рекомендации по организации и формы проведения лекционных занятий по учебной дисциплине «Языки программирования»

№	Темы лекционных занятий (форма проведения)	Трудоем- кость в АЧ
1	Общее знакомство. Происхождение языка Си (информационная лекция)	2
2	Знакомство с указателями, классы памяти и разработка программ (информационная лекция)	2
3	Использование текстового редактора для подготовки программ (информационная лекция)	2
4	Примеры простой программы. Структура простой программы (информационная лекция)	2
5	Структурное программирование (информационная лекция)	2
6	Библиотеки языка Си (информационная лекция)	2
7	Функции и макроопределения, распределение памяти, примеры функций (информационная лекция)	2
8	Массивы и указатели, символьные строки и функции над ними. Структуры (информационная лекция)	2
9	Циклы и другие управляющие средства, создание и использование функций (информационная лекция)	2
	ИТОГО	18

Таблица 6 – Методические рекомендации по организации для очной формы обучения проведения практических занятий по учебной дисциплине «Языки программирования»

№	Темы практических занятий (форма проведения)	Трудоем- кость в АЧ
1	Простейшие вычисления с вводом/выводом данных на консоль (опрос)	6
2	Составьте программу, которая подсчитывает и выводит значение $t1$ и $t2$ по формулам, которые приведены в Вашем варианте индивидуального задания. Значения параметров с именами x и y должны вводиться с клавиатуры и имеют тип <code>double</code> . Допускается упростить / разложить формулы для того, чтобы обеспечить минимизацию объема вычислений. (опрос)	6
3	Написать программу, вычисляющую и печатающую значение переменных $t1$ и $t2$ по формуле, соответствующей вашему варианту. Параметры с именами m и n имеют тип <code>int</code> . Остальные параметры – с плавающей точкой (<code>double</code>). Значения параметров x и y вводятся с клавиатуры, значения остальных параметров – задаются при объявлении соответствующих переменных. (опрос)	6
4	Расчет координат точек на плоскости (опрос)	6
5	Интерационные вычисления (опрос)	6
6	Обработка массивов данных (опрос)	6
	ИТОГО	36

Таблица 7 – Методические рекомендации по организации для заочной формы обучения формы (11.03.03 Конструирование и технология электронных средств) проведения практических занятий по учебной дисциплине «Языки программирования»

№	Темы практических занятий (форма проведения)	Трудоем- кость в АЧ
1.	Простейшие вычисления с вводом/выводом данных на консоль (опрос)	2
2.	Составьте программу, которая подсчитывает и выводит значение $t1$ и $t2$ по формулам, которые приведены в Вашем варианте индивидуального задания. Значения параметров с именами x и y должны вводиться с клавиатуры и имеют тип <code>double</code> . Допускается упростить / разложить формулы для того, чтобы обеспечить минимизацию объема вычислений. (опрос)	2
3.	Написать программу, вычисляющую и печатающую значение переменных $t1$ и $t2$ по формуле, соответствующей вашему варианту. Параметры с именами m и n имеют тип <code>int</code> . Остальные параметры – с плавающей точкой (<code>double</code>). Значения параметров x и y вводятся с клавиатуры, значения остальных параметров – задаются при объявлении соответствующих переменных. (опрос)	2
4.	Итерационные вычисления (опрос)	2
	ИТОГО	8

6 Фонд оценочных средств учебной дисциплины «Языки программирования»

Фонд оценочных средств представлен в Приложении А.

7 Условия освоения учебной дисциплины

Лекционные занятия проводятся с применением демонстрационного материала в компьютерном классе с мультимедийным оборудованием и компьютерами (ауд. 2701).

Практические занятия проводятся в компьютерном классе с мультимедийным оборудованием и компьютерами (ауд. 2701). На компьютерах установлен лицензионный пакет программ «Word, Excel, Paint, PowerPoint, Access» для визуального схемотехнического моделирования, а также ряд программ со свободной лицензией

Самостоятельная работа студентов (аудиторная и внеаудиторная) осуществляется с применением компьютеров с выходом в сеть «Интернет» для изучения рекомендованных ресурсов и самостоятельного поиска информации.

7.1 Учебно-методическое обеспечение

Карта учебно-методического обеспечения учебной дисциплины (модуля) представлено в Приложении Б.

7.2 Материально-техническое обеспечение

Таблица 8 – Материально-техническое обеспечение учебной дисциплины «Языки программирования»

№	Требование к материально-техническому обеспечению	Наличие материально-технического оборудования и программного обеспечения
1.	Наличие специальной аудитории	Компьютерный класс ПК Intel Celeron G620 CPU 2,6 GHz, Монитор Samsung S20B300 – 8 шт.
2.	Мультимедийное оборудование	Мультимедийная система: ПК Intel Celeron G630 CPU 2,7 GHz, Монитор XEROX XA7-17, Проектор Epson Projector EB-X11 (лицензия Windows 7 Professional – MDK37-BGF99-8CWKQ-T7KGD-9DJG9)
3.	Программное обеспечение	Word, Excel, Paint, PowerPoint, Access (лицензия Microsoft Office № 62846215ZZE0910), Adobe (лицензия № 65014264ES), Среда разработки NI LabVIEW 2012 (лицензия № M77X11147)

Приложение А
(обязательное)
Фонд оценочных средств
учебной дисциплины «Языки программирования»

1. Структура фонда оценочных средств

Фонд оценочных средств состоит из двух частей:

а) открытая часть - общая информация об оценочных средствах (название оценочных средств, проверяемые компетенции, баллы, количество вариантов заданий, методические рекомендации для применения оценочных средств и пр.), которая представлена в данном документе, а также те вопросы и задания, которые могут быть доступны для обучающегося;

б) закрытая часть - фонд вопросов и заданий, которая не может быть заранее доступна для обучающихся (экзаменационные билеты, вопросы к контрольной работе и пр.) и которая хранится на кафедре.

2. Перечень оценочных средств текущего контроля и форм промежуточной аттестации

Таблица А.1 – Перечень оценочных средств

<i>№</i>	<i>Оценочные средства для текущего контроля</i>	<i>Разделы учебной дисциплины</i>	<i>Баллы</i>	<i>Проверяемые компетенции</i>
1	Опрос, практические задания	1÷9 (в таблице 3)	150	ОПК-3 ОПК-4
<i>Промежуточная аттестация</i>				
	Зачет			ОПК-3 ОПК-4
	ИТОГО		150	

3. Рекомендации к использованию оценочных средств

Таблица А.2 - Контрольный опрос

<i>Критерии оценки</i>	<i>Количество вариантов в заданиях</i>	<i>Количество вопросов</i>
демонстрирует всестороннее и глубокое знание теоретического материала и задача решена правильно	По числу обучающихся	2
допускает неточности при ответе на теоретические вопросы и (или) допущены неточности в решении задачи		
испытывает трудности при ответе на теоретические вопросы или задача решена неправильно		

Примерные вопросы:

1. История и назначение языка Си++.
2. Структура программы на языке С++. Этапы создания исполняемой программы.
3. Состав языка С++. Константы и переменные С++.
4. Типы данных в С++.
5. Выражения. Знаки операций. Постфиксные и префиксные операции
6. Массивы (определение, инициализация, способы перебора).
7. Сортировка массивов (простой обмен, простое включение, простой выбор).
8. Указатели. Операции с указателями.
9. Одномерные массивы и указатели.
10. Многомерные массивы и указатели.
11. Символьная информация и строки. Функции для работы со строками (библиотечный файл string.h).
12. Функции в С++. Рекурсия. Примеры.
13. Прототип функции. Библиотечные файлы. Директива препроцессора #include.
14. Передача одномерных массивов в функции.
15. Передача многомерных массивов в функции.
16. Передача строк в функции.
17. Подставляемые функции.
18. Ссылки на функции.
19. Генерация заданного количества цифр числа пи.
20. Работа со ссылками. Регулярные выражения. Основные встроенные функции

Таблица А.3 - Практические задания

<i>Критерии оценки</i>	<i>Количество вариантов заданий</i>
способен правильно проанализировать нужные материалы и правильно их применить	10
способен правильно проанализировать нужные материалы, но допускает некритические ошибки в их использовании	
не всегда адекватно анализирует материалы для решения задач и (или) использует их с ошибками	

Практические занятия:

- Программный комплекс "МВТУ" (Моделирование в технических устройствах);
- Создание функциональной модели с помощью ВРwin;
- Комплексные решения на основе системы ОРГ-МАСТЕР®;
- Среда прикладного графического программирования LabVIEW;
- Обработка данных эксперимента с использованием программно-методического комплекса Statistica.

Содержание отчета

1. Титульный лист
2. Правильный результат выполнения тестового примера корректности
3. Отчет оформлен.
4. Обоснованный ответ на вопросы по темам ПЗ:
 - описание алгоритма
 - схема алгоритма
 - листинг программ
 - описание тестового примера и результата его выполнения
 - отчет оформлен по ГОСТ 2.105-98

Все материалы для проведения промежуточного контроля хранятся на кафедре.

Приложение Б
(обязательное)
Карта учебно-методического обеспечения
Учебной дисциплины «Языки программирования»

Таблица Б.1- Основная литература*

Библиографическое описание издания (автор, наименование, вид, место и год издания, кол. стр.)	Кол. экз. в библ. НовГУ	Наличие в ЭБС
Печатные источники		
Немнюгин С. А., Перколаб Л.В. Изучаем Turbo Pascal:- СПб.:Питер,2005.-309с.	12	
Труб И. И., Объектно-ориентированное моделирование на C++:Учеб. курс-СПб.:Питер,2006.-410с.	13	
Опалева Э. А., Самойленко В.П. Языки программирования и методы трансляции: учеб. пособие-СПб.:БХВ-Петербург,2005.- 476с.	30	
Кузнецов А., Microsoft Access 2003. Русская версия:учеб. курс- СПб.:Питер,2006.-364с.	10	
Павловская Т. А., Щупак Ю.А. С/С++. Структурное программирование:практикум-СПб.:Питер,2007.-238с.	18	
Немнюгин С. А., Turbo Pascal. Программирование на языке высокого уровня:учеб. пособие для вузов-СПб.:ПИТЕР,2006.- 543с.	13	
Храмцов П. Б., Брик С.А. Основы web-технологий:учеб. пособие для вузов-М.:Интернет-Университет Информ. Технологий,2007.-374с.	20	
Каймин В. А., Информатика:учеб. для вузов по естеств.-науч. направл. и спец.-М.:Инфра-М,2008.-283,[2]с.	11	
Баженова И. Ю., Сухомлин В.А. Введение в программирование:учеб. пособие-М.:Интернет-Университет Информ. Технологий,2007.-326с.	12	
Павловская Т. А., Паскаль. Программирование на языке высокого уровня:учеб. для вузов-СПб.:Питер,2006.-392с.	10	
Лаптев В. В., С++. Объектно-ориентированное программирование:учеб. пособие-СПб.:Питер,2008.-457с.	10	
Фаронов В. В., Delphi. Программирование на языке высокого уровня:учеб. для вузов-СПб.:Питер,2009.-639,[1]с.	10	
Бобровский С. И., Delphi 7:учеб. курс-СПб.:Питер,2008.-735с.	10	
Костюкова Н. И., Калинина Н.А. Язык Си и особенности работы с ним:учеб. пособие-М.:Интернет-Университет Информ. Технологий,2006.-205с.	20	
Давыдова Н.А., Боровская Е.В. Программирование:учеб. пособие-М.:БИНОМ. Лаборатория знаний,2009.-238,[1]с.	10	
Функциональное и логическое программирование : лаб. практикум. Ч. 2 : Логическое программирование / сост.: Д. В. Михайлов, Г. М. Емельянов; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2008. - 105с.	11	

Электронные ресурсы		
Применение логических языков программирования для решения задач искусственного интеллекта : учеб. пособие / А. В. Колногоров, Д. Н. Шиян ; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2015. - 87 с.	16	https://novsu.bibliotech.ru/Reader/Book/-2627
Гарбарь С. В. Моделирование стационарных случайных последовательностей с использованием алгоритмов авторегрессии и их модификаций : монография / С. В. Гарбарь; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2016. - 76 с.	12	https://novsu.bibliotech.ru/Reader/Book/-2833
Функциональное и логическое программирование : лаб. практикум. Ч. 1 : Функциональное программирование / сост.: Д. В. Михайлов, Г. М. Емельянов ; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2007. - 80 с	11	https://novsu.bibliotech.ru/Reader/Book/-432

Таблица Б.2 - Дополнительная литература

Библиографическое описание издания (автор, наименование, вид, место и год издания, кол. стр.)	Кол. экз. в библ. НовГУ	Наличие в ЭБС
Печатные источники		
Макарова Н. В., Култышев Е.И. Информатика. Практикум по технологии работы на компьютере: учеб. пособие для студентов вузов-М.: Финансы и статистика,2004.-255с.	26	
Аскеров Т. М., Данчул А.Н. Информатика:учебник-М.:Издательство РАГС,2004.-525с.	15	
Павловская Т. А., С/С++. Программирование на языке высокого уровня:учеб. для вузов-СПб.:Питер,2006.-460с.	21	
Солонина А.И., Улахович Д.А. Алгоритмы и процессоры цифровой обработки сигналов:учеб. пособие для вузов-СПб.:БХВ-С.-Петербург,2001.-454с.	5	
Электронные ресурсы		
Язык программирования Паскаль : учеб.-метод. пособие / авт.-сост.: Т. А. Митенева, М. Л. Федорова ; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2004. - 49 с.	9	https://novsu.bibliotech.ru/Reader/Book/-452

Заведующий кафедрой РС _____ И.Н.Жукова
« _____ » _____ 20__ г.

Приложение В
(обязательное)

**Лист актуализации рабочей программы
учебной дисциплины (модуля) «Языки программирования»**

Рабочая программа актуализирована на 20__/20__ учебный год.
Протокол № ____ заседания кафедры от « ____ » _____ 20__ г.
Разработчик: _____
Зав. кафедрой _____

Рабочая программа актуализирована на 20__/20__ учебный год.
Протокол № ____ заседания кафедры от « ____ » _____ 20__ г.
Разработчик: _____
Зав. кафедрой _____

Рабочая программа актуализирована на 20__/20__ учебный год.
Протокол № ____ заседания кафедры от « ____ » _____ 20__ г.
Разработчик: _____
Зав. кафедрой _____

Перечень изменений, внесенных в рабочую программу:

Номер изменения	№ и дата протокола заседания кафедры	Содержание изменений	Зав.кафедрой	Подпись

Закрытая часть для очной формы обучения
учебной дисциплины
«Языки программирования»
ФГОС 3++
Варианты практических занятий

№1

№	Задание
1	Даны переменные <code>int a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a + b</code> обратно в консоль, используя шестнадцатеричное представление результата.
2	Даны переменные <code>float a, z</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a/z</code> обратно в консоль. Для результата задать количество знаков после запятой равное 3.
3	Даны переменные <code>unsigned int a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a%z</code> обратно в консоль. Для результата использовать шестнадцатеричное представление вывода.
4	Даны переменные <code>float a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a/b</code> обратно в консоль. Для результата использовать экспоненциальный формат представления.
5	Даны переменные <code>int x, y</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>x*y</code> обратно в консоль. Для результата использовать восьмеричный формат представления
6	Даны переменные <code>float a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a-b</code> обратно в консоль. Для результата задать количество знаков после точки равное 5.
7	Даны переменные <code>short a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a*k</code> обратно в консоль. Для результата использовать восьмеричную форму представления.
8	Даны переменные <code>double a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>2*a+k</code> обратно в консоль. Для результата использовать экспоненциальную форму представления.
9	Даны переменные <code>short x, y</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>x/(1-y)</code> обратно в консоль. Для результата использовать десятичный формат представления
10	Даны переменные <code>float a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a/(b-7)</code> обратно в консоль. Для результата задать количество знаков до точки равное 5.

№2

Составьте программу, которая подсчитывает и выводит значение $t1$ и $t2$ по формулам, которые приведены в Вашем варианте индивидуального задания. Значения параметров с именами x и y должны вводиться с клавиатуры и имеют тип `double`. Допускается упростить / разложить формулы для того, чтобы обеспечить минимизацию объема вычислений.

Варианты практических заданий

1	$t1 = -x + \frac{\sqrt{x}}{\sin(y^3)}$	$t2 = \frac{1}{y} \ln(3x)$
2	$t1 = \frac{\sqrt{3xy}}{12 - xy} - \frac{2}{y^2}$	$t2 = -\frac{x}{2} \left(\frac{\ln(x)}{y} \right)$
3	$t1 = \frac{1}{5x} \ln \left(\frac{2-y}{x-2} \right)$	$t2 = \frac{3x + \cos(y)}{3x^3}$
4	$t1 = \frac{y^2}{3} + \ln(2x)$	$t2 = \frac{3 + \sin(y)}{\cos(x)}$
5	$t1 = -\frac{y}{2} \ln \frac{1}{\sqrt{2x}}$	$t2 = \frac{\sin(5+x)}{2\sin^2(3y-4)}$
6	$t1 = \frac{1}{x} \ln \left(\frac{y}{\sqrt{y^2 - x^2}} \right)$	$t2 = \frac{1}{4 + \sqrt{x}} + \frac{2\cos^2(2x)}{\sin^2(y)}$
7	$t1 = \frac{x}{3} - \frac{\sin(2x^3)}{y^5}$	$t2 = \frac{y + \operatorname{ctg}(x)}{2y}$
8	$t1 = \frac{\sin(5+y)}{x^{-2}} - \ln(x)$	$t2 = \frac{7 \sin(x)}{x} \operatorname{tg}(y/x)$
9	$t1 = \frac{y}{6-x} \ln \left(\frac{4x}{y} \right)$	$t2 = \frac{1}{y} \operatorname{ctg}(3x^3)$
10	$t1 = \frac{1}{\sin^2(y)} 3\sqrt{x^2 - 2}$	$t2 = \frac{1}{3x} \lg \left(\frac{y}{2-x} \right)$

1	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Поле управления диспетчером кадра в протоколе управления логическим каналом имеет формат: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>1</td><td>0</td><td>C</td><td>C</td><td>C</td><td>A</td><td>A</td><td>A</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>P</td></tr></table> Где: C..C – управляющий код A..A – идентификатор команды N..N – номер последовательности P – признак запроса или завершения (0/1)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	1	0	C	C	C	A	A	A	N	N	N	N	N	N	N	P
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	1	0	C	C	C	A	A	A	N	N	N	N	N	N	N	P																			
2	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Информация о критической ошибке на диске представляется в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>O</td><td>A</td><td>A</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>X</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td></tr></table> Где: O – тип операции: чтение (0) / запись (1) AA – код области диска R..R – код операции, выполняемой при ошибке X – признак защищенной области диска D..D – номер диска	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	O	A	A	R	R	R	R	R	X	D	D	D	D	D	D	D
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	O	A	A	R	R	R	R	R	X	D	D	D	D	D	D	D																			
3	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Дескриптор сегмента для системы виртуальной памяти представляется в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>W</td><td>R</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td></tr></table> Где: L..L – размер сегмента в блоках F..F – номер блока, с которого начинается сегмент R – флаг доступности сегмента для чтения (0/1) W – флаг доступности сегмента для записи (0/1)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	W	R	F	F	F	F	F	F	F	L	L	L	L	L	L	L
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	W	R	F	F	F	F	F	F	F	L	L	L	L	L	L	L																			

4	Составьте программу, которая делает «переключение» битов (toggle) во введенном пользователем числе m (тип unsigned char). Начальную позицию и количество переключаемых битов (идущих друг за другом) вводится пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе использовать представление в 16 системе счисления. Пример: Введенное число: 0xae (10101110) Позиция 4 (индексы битов начинаются с 0) и количество - 3 Результат: 0xde (11011110)																																		
5	Составьте программу, которая циклически сдвигает введенное пользователем число m (тип unsigned int) на n разрядов. Если n > 0 – сдвиг вправо, если n < 0 – сдвиг влево. Значения m и n вводятся пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе m использовать представление в 16-ой системе счисления. Пример: Введенное число: 0x3 (0000 0000 0000 0000 0000 0000 0000 0011) Введенное значение n = 3 Результат: 0x60000000 (0110 0000 0000 0000 0000 0000 0000 0000)																																		
6	Составьте программу, которая делает «замену» битов во введенном пользователем числе m (тип unsigned char). Позицию замены и новое значение заменяемого бита вводится пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе использовать представление в 16 системе счисления. Пример: Введенное число: 0xae (10101110) Позиция 4 (индексы битов начинаются с 0) и новое значение - 1 Результат: 0xbe (10111110)																																		
7	С клавиатуры вводится целое число A (тип unsigned char) , нужно вывести на экран младшие 4 бита и старшие 4 бита. Затем в числе A произвести перестановку местами двух старших и двух младших бит и вывести полученное значение на экран. Пример: Введенное число: A = 229 (1110 0101) Вывод программы: Старшие биты: 14, младшие биты: 5, результат перестановки бит: 103 (01100111)																																		
8	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Идентификатор мобильного телефона для станции представлен в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>P</td><td>P</td><td>P</td><td>X</td><td>B</td><td>B</td><td>B</td><td>B</td><td>V</td><td>E</td><td>E</td></tr></table> Где: PPP – мощность сигнала A...A – рабочая группа устройства X – признак блокировки устройства BBB – уровень заряда батареи V – признак поддержки видеосвязи EE – тип сотовой связи (GSM/EDGE/3G/LTE)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	A	A	A	A	A	P	P	P	X	B	B	B	B	V	E	E
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	A	A	A	A	A	P	P	P	X	B	B	B	B	V	E	E																			

9

Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип **unsigned int**), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран.

Слово состояния канала в вычислительной системе представляется в виде:

№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Значение	N	N	N	N	E	K	K	C	C	C	C	C	C	C	0	P

Где:

N..N - номер канала

E - признак ошибки

KK – код статуса канала

C..C - код причины прерывания

P - признак завершения программы в канале

10

Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип **unsigned int**), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран.

Элемент списка безопасности объекта в вычислительной системе представляется в виде:

№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Значение	G	G	G	G	G	G	U	U	U	U	U	U	0	R	W	X

Где:

G..G – идентификатор группы пользователя

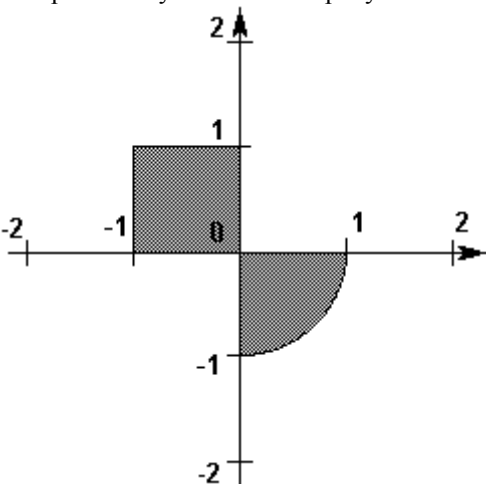
U..U – идентификатор пользователя

R – флаг, определяющий право чтения

W – флаг, определяющий право записи

X – флаг, определяющий право выполнения программного кода

№4

1	Пользователь вводит с клавиатуры координаты 4-х точек: A (x1, y1) и B (x2, y2) , C (x3, y3) и D(x4, y4) , задающих два отрезка. Написать программу, которая определяет: середина какого отрезка лежит ближе всего к началу координат
2	Пользователь вводит с клавиатуры координаты трех точек – вершин треугольника: A (x1, y1) , B (x2, y2) , C (x3, y3) . Написать программу, которая находит длину наименьшей стороны данного треугольника.
3	Написать программу, которая вводит координаты точки (x, y) (тип double) и определяет, попадает ли точка в заштрихованную область на рисунке. Попадание на границу области считать попаданием в область. 
4	Пользователь вводит с клавиатуры координаты трех точек: A (x1, y1) , B (x2, y2) , C (x3, y3) . Составьте программу, выводящую на экран координаты той точки, которая расположена ближе всех к центру координат (0, 0), и длину до него. Переменные, хранящие координаты точек имеют тип double .

1.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 15 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \left(1 - \frac{(n+1)^2}{(n+2)^2}\right)$
2.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 30 членов. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{n+1}{n^3+1}$
3.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 в зависимости от переменной x (тип double). Значение переменной вводится пользователем с клавиатуры. $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + \frac{(-1)^n x^{n+1}}{n+1} \pm \dots, \quad -1 < x \leq 1$
4.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 15 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{n^2+1}{n^3+3}$
5.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 в зависимости от переменной x и a (тип double). Значения переменных вводятся пользователем с клавиатуры. $a^x = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \dots + \frac{(x \ln a)^n}{n!} + \dots$
6.	Составьте программу, вычисляющую значение синуса угла (в градусах) по разложению в ряд Маклорена, приведенного ниже. Переменная X вводится пользователем с клавиатуры и имеет тип double. Результат вычислений выведите с точностью до 6-го знака после запятой. $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n x^{2n+1}}{(2n+1)!} + \dots$
7.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 20 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{n+1}{3^n+2^n}$
8.	Составьте программу, вычисляющую число π через ряд Лейбница, приведенный ниже, до n = 10 000 000. На каждой миллионной итерации выводите промежуточный результат вычисления.

	Результаты вычислений выводите с точностью до 10 знака после запятой. $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \frac{1}{17} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$
9.	Составьте программу, вычисляющую значение косинуса введенного пользователем угла (в градусах) по разложению в ряд Маклорена, приведенного ниже. Переменная X вводится пользователем с клавиатуры и имеет тип double. Результат вычислений выведите с точностью до 6-го знака после запятой. $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + \frac{(-1)^n x^{2n}}{(2n)!} + \dots$
10.	Составьте программу, вычисляющую приближенное значение числа Эйлера (e) по разложению в ряд Маклорена. Переменная X вводится пользователем с клавиатуры и имеет тип double. Результат вычислений выведите с точностью до 6-го знака после запятой: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!} + \dots$
11.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 15 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{n^2 + 1}{n^3 + 3}$
12.	Составьте программу, вычисляющую значение синуса угла (в градусах) по разложению в ряд Маклорена, приведенного ниже. Переменная X вводится пользователем с клавиатуры и имеет тип double. Результат вычислений выведите с точностью до 6-го знака после запятой. $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n x^{2n+1}}{(2n+1)!} + \dots$
13.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 25 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{2^n}{n^{n+1} + 1}$
14.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 10 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{2^{n+1}}{2^{2n} + 1}$
15.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 30 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1)^n \frac{1}{n^2 + 2^n}$

1	Написать программу, которая находит второй наибольший элемент массива размером N. Размер массива вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран полученный массив и значение найденного элемента. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Второй наибольший элемент: 38
2	Написать программу, которая находит количество четных и нечетных элементов в массиве размером N. Размер массива вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран полученный массив и найденные значения. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Количество четных элементов: 6 Количество нечетных элементов: 4
3	Написать программу, которая циклически переставляет элементы массива размером N на заданное количество шагов. Размер массива вводится пользователем с клавиатуры. Количество шагов также вводится с клавиатуры, причем положительное значение соответствует перестановке по часовой стрелке (вправо), отрицательно — против часовой стрелки (влево). Для заполнения массива использовать функцию rand. Программа должна вывести на экран исходный массив и массива полученный в результате перестановки. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Количество шагов: 5 Результат выполнения программы: 6 75 38 3 2 -7 2 3 8 6
4	Написать программу, которая находит количество всех повторяющихся элементов в массиве размером N. Размер массива вводится пользователем с клавиатуры. Элементы массива также вводятся пользователем с клавиатуры. Программа должна вывести на экран полученный массив и найденное значение. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Количество повторяющихся элементов: 3
5	Написать программу, которая находит среднее арифметическое всех четных и всех нечетных элементов в массиве размером N. Размер массива вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран полученный массив и найденные значения. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Среднее арифметическое четных элементов: 10.3 Среднее арифметическое нечетных элементов: 18.5
6	Написать программу, которая находит количество все уникальные (не повторяющиеся) элементы в массиве размером N. Размер массива вводится пользователем с клавиатуры. Элементы массива также вводятся пользователем с клавиатуры. Программа должна вывести на экран исходный массив и список уникальных значений. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Уникальные элементы: -7 8 75 38
7	Написать программу, которая меняет местами элементы на четных и нечетных местах в массиве размером N. Размер массива вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран исходный массив и обработанный массив. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Обработанный массив: 2 -7 8 3 6 6 38 75 2 3
8	Написать программу, которая находит среднее арифметическое всех элементов в массиве размером N, а затем обнуляет все элементы, которые больше данного значения. Размер массива вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран исходный массив и найденное среднее арифметическое и обработанный массив.

	Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Среднее арифметическое: 13.6 Обработанный массив: -7 2 3 8 6 6 0 0 3 2
9	Написать программу, которая выполняет поиск заданного элемента в массиве размером N. Размер массива и значение искомого элемента вводится пользователем с клавиатуры. Поиск прекращается при для первого найденного элемента. Для заполнения массива использовать функцию rand. Программа должна вывести на экран исходный массив и индекс найденного элемента, если элемент не найден — индекс равен -1. Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Искомый элемент: 6 Результат: 4
10	Написать программу, которая выполняет вставку элемента в массиве размером N на указанную позицию (индекс). Размер массива, значение и индекс вставляемого элемента вводится пользователем с клавиатуры. Для заполнения массива использовать функцию rand. Программа должна вывести на экран исходный массив и результирующий массив (размером N+1). Пример: Исходный массив: -7 2 3 8 6 6 75 38 3 2 Вставляемый элемент: 9, позиция вставки 5 Результат: -7 2 3 8 6 9 6 75 38 3 2

Закрытая часть для заочной формы обучения
11.03.03 Конструирование и технология электронных средств
учебной дисциплины
«Языки программирования»
ФГОС 3++
Варианты практических занятий

№1

№	Задание
1	Даны переменные <code>int a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a + b</code> обратно в консоль, используя шестнадцатеричное представление результата.
2	Даны переменные <code>float a, z</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a/z</code> обратно в консоль. Для результата задать количество знаков после запятой равное 3.
3	Даны переменные <code>unsigned int a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a%z</code> обратно в консоль. Для результата использовать шестнадцатеричное представление вывода.
4	Даны переменные <code>float a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a/b</code> обратно в консоль. Для результата использовать экспоненциальный формат представления.
5	Даны переменные <code>int x, y</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>x*y</code> обратно в консоль. Для результата использовать восьмеричный формат представления
6	Даны переменные <code>float a, b</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a-b</code> обратно в консоль. Для результата задать количество знаков после точки равное 5.
7	Даны переменные <code>short a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>a*k</code> обратно в консоль. Для результата использовать восьмеричную форму представления.
8	Даны переменные <code>double a, k</code> . Получить из значения из консоли, а затем вывести результат вычисления выражения <code>2*a+k</code> обратно в консоль. Для результата использовать экспоненциальную форму представления.

1	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Поле управления диспетчером кадра в протоколе управления логическим каналом имеет формат: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>1</td><td>0</td><td>C</td><td>C</td><td>C</td><td>A</td><td>A</td><td>A</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>P</td></tr></table> Где: C..C – управляющий код A..A – идентификатор команды N..N – номер последовательности P – признак запроса или завершения (0/1)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	1	0	C	C	C	A	A	A	N	N	N	N	N	N	N	P
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	1	0	C	C	C	A	A	A	N	N	N	N	N	N	N	P																			
2	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Информация о критической ошибке на диске представляется в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>O</td><td>A</td><td>A</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>X</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td></tr></table> Где: O – тип операции: чтение (0) / запись (1) AA – код области диска R..R – код операции, выполняемой при ошибке X – признак защищенной области диска D..D – номер диска	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	O	A	A	R	R	R	R	R	X	D	D	D	D	D	D	D
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	O	A	A	R	R	R	R	R	X	D	D	D	D	D	D	D																			
3	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Дескриптор сегмента для системы виртуальной памяти представляется в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>W</td><td>R</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td></tr></table> Где: L..L – размер сегмента в блоках F..F – номер блока, с которого начинается сегмент R – флаг доступности сегмента для чтения (0/1) W – флаг доступности сегмента для записи (0/1)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	W	R	F	F	F	F	F	F	F	L	L	L	L	L	L	L
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	W	R	F	F	F	F	F	F	F	L	L	L	L	L	L	L																			

4	Составьте программу, которая делает «переключение» битов (toggle) во введенном пользователем числе m (тип unsigned char). Начальную позицию и количество переключаемых битов (идущих друг за другом) вводится пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе использовать представление в 16 системе счисления. Пример: Введенное число: 0xae (10101110) Позиция 4 (индексы битов начинаются с 0) и количество - 3 Результат: 0xde (11011110)																																		
5	Составьте программу, которая циклически сдвигает введенное пользователем число m (тип unsigned int) на n разрядов. Если n > 0 – сдвиг вправо, если n < 0 – сдвиг влево. Значения m и n вводятся пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе m использовать представление в 16-ой системе счисления. Пример: Введенное число: 0x3 (0000 0000 0000 0000 0000 0000 0000 0011) Введенное значение n = 3 Результат: 0x60000000 (0110 0000 0000 0000 0000 0000 0000 0000)																																		
6	Составьте программу, которая делает «замену» битов во введенном пользователем числе m (тип unsigned char). Позицию замены и новое значение заменяемого бита вводится пользователем с клавиатуры. Выведите полученное число на экран. При вводе и выводе использовать представление в 16 системе счисления. Пример: Введенное число: 0xae (10101110) Позиция 4 (индексы битов начинаются с 0) и новое значение - 1 Результат: 0xbe (10111110)																																		
7	С клавиатуры вводится целое число A (тип unsigned char) , нужно вывести на экран младшие 4 бита и старшие 4 бита. Затем в числе A произвести перестановку местами двух старших и двух младших бит и вывести полученное значение на экран. Пример: Введенное число: A = 229 (1110 0101) Вывод программы: Старшие биты: 14, младшие биты: 5, результат перестановки бит: 103 (01100111)																																		
8	Написать программу, состоящую из двух частей, первая из которых читает из консоли составные части структуры данных, приведенной ниже, и формирует из них заданную упакованную структуру (тип unsigned int), которую выводит на экран в 16-ном формате представления. Вторая - читает упакованную структуру как 16-ричное число и выводит значения отдельных ее составных частей на экран. Идентификатор мобильного телефона для станции представлен в виде: <table><tr><td>№ разряда</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Значение</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>P</td><td>P</td><td>P</td><td>X</td><td>B</td><td>B</td><td>B</td><td>B</td><td>V</td><td>E</td><td>E</td></tr></table> Где: PPP – мощность сигнала A...A – рабочая группа устройства X – признак блокировки устройства BBB – уровень заряда батареи V – признак поддержки видеосвязи EE – тип сотовой связи (GSM/EDGE/3G/LTE)	№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Значение	A	A	A	A	A	P	P	P	X	B	B	B	B	V	E	E
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Значение	A	A	A	A	A	P	P	P	X	B	B	B	B	V	E	E																			

16.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 15 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1^n) \left(1 - \frac{(n+1)^2}{(n+2)^2}\right)$
17.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 30 членов. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1^n) \frac{n+1}{n^3+1}$
18.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 в зависимости от переменной x (тип double). Значение переменной вводится пользователем с клавиатуры.. $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + \frac{(-1)^n x^{n+1}}{n+1} \pm \dots, \quad -1 < x \leq 1$
19.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 15 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1^n) \frac{n^2+1}{n^3+3}$
20.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 в зависимости от переменной x и a (тип double). Значения переменных вводятся пользователем с клавиатуры. $a^x = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \dots + \frac{(x \ln a)^n}{n!} + \dots$
21.	Составьте программу, вычисляющую значение синуса угла (в градусах) по разложению в ряд Маклорена, приведенного ниже. Переменная X вводится пользователем с клавиатуры и имеет тип double. Результат вычислений выведите с точностью до 6-го знака после запятой. $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n x^{2n+1}}{(2n+1)!} + \dots$
22.	Для ряда, члены которого вычисляются по формуле ниже, подсчитать сумму членов ряда с точностью до 0.000001 и сумму первых 20 членов ряда. Если Вы считаете это необходимым, можете упростить или преобразовать выражение. $a_n = (-1^n) \frac{n+1}{3^n+2^n}$
23.	Составьте программу, вычисляющую число π через ряд Лейбница, приведенный ниже, до n = 10 000 000. На каждой миллионной итерации выводите промежуточный результат вычисления. Результаты вычислений выводите с точностью до 10 знака после запятой. $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \frac{1}{17} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$