

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

“УТВЕРЖДАЮ”

Декан АВТФ

профессор, д.т.н. Гужов
Владимир Иванович

“ ___ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дополнительные главы информатики

ООП: специальность 090105.65 Комплексное обеспечение информационной безопасности автоматизированных систем

Шифр по учебному плану: ЕН.В.2.1

Факультет: автоматики и вычислительной техники очная форма обучения

Курс: 1, семестр: 2

Лекции: 34

Практические работы: 16 Лабораторные работы: -

Курсовой проект: - Курсовая работа: - РГЗ: -

Самостоятельная работа: 30

Экзамен: - Зачет: 2

Всего: 80

Новосибирск

2011

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования по направлению (специальности): 075500 Комплексное обеспечение информационной безопасности автоматизированных систем.(№ 284 инф/сп от 05.04.2000)

ЕН.В.2.1, дисциплины по выбору студента

Рабочая программа обсуждена на заседании кафедры Вычислительной техники протокол № 1 от 28.01.2011

Программу разработал

профессор, д.т.н.

Гужов Владимир Иванович

Заведующий кафедрой

профессор, д.т.н.

Губарев Василий Васильевич

Ответственный за основную образовательную программу

с.н.с., к.т.н.

Трушин Виктор Александрович

1. Внешние требования

Таблица 1.1

Шифр дисциплины	Содержание учебной дисциплины	Часы
ЕН.В.2.1	<p>Рабочая программа составлена на основании государственного образовательного стандарта высшего профессионального образования по специальности 075500 (090105) "Комплексное обеспечение информационной безопасности автоматизированных систем" (квалификация - специалист по защите информации), утвержденного приказами Минобрнауки России №2719 от 30.12.1997 г., № 686 от 2.03.2000 г. и №1010 от 6.04.2000 г.</p> <p>Дисциплина относится к блоку "Общие, математические и естественнонаучные дисциплины" (ЕН.Ф.03).</p> <p>Ниже приведены выдержки из первого раздела ГОС "ОБЩАЯ ХАРАКТЕРИСТИКА СПЕЦИАЛЬНОСТИ":</p> <p>"Объектами профессиональной деятельности выпускника являются автоматизированные системы обработки, хранения и передачи информации определенного уровня конфиденциальности, методы и средства обеспечения информационной безопасности автоматизированных систем."</p> <p>Содержание дисциплины "Дополнительные главы информатика" является продолжением курса "Информатика".</p> <p>4. ТРЕБОВАНИЯ К ОБЯЗАТЕЛЬНОМУ МИНИМУМУ СОДЕРЖАНИЯ</p> <p>ЕН.Ф.03 Информатика 100</p> <p>понятие информации; общая характеристика процесса сбора, передачи, обработки и накопления информации; понятие алгоритма; ЭВМ как исполнитель алгоритмов: структура вычислительных машин, принципы их функционирования, основные характеристики, история развития; структура программы на языке высокого уровня, представление текста программы, оформление программы; развитие языков программирования; этапы разработки программ; основные типы ЭВМ: архитектура, назначение центрального процессора и основной памяти, классы памяти, типы и назначение внешних устройств; представление данных в памяти ЭВМ; обработка данных в центральном процессоре: типы команд, реализация операций и операторов языка высокого уровня на языке ассемблера; особенности организации персональных ЭВМ; организация интерфейса взаимодействия с пользователями и прикладными программами; организация эффективной работы с пакетами</p>	80

	<p>прикладных программ; понятие системного программного обеспечения: назначение, возможности, структура; операционные системы для различных ЭВМ: файловая система, система управления работой пользователей, командные языки; трансляторы и редакторы связей; этапы работы на ЭВМ; особенности работы на ПЭВМ; основные понятия систем управления базами данных и базами знаний.</p>	
--	--	--

2. Особенности (принципы) построения дисциплины

Таблица 2.1

Особенности (принципы) построения дисциплины

Особенность (принцип)	Содержание
Основания для введения дисциплины в учебный план по направлению или специальности	<p>Рабочая программа составлена на основании государственного образовательного стандарта высшего профессионального образования по специальности: 075500 (090105) - Комплексное обеспечение информационной безопасности автоматизированных систем.</p> <p>Дисциплина относится к блоку "Общие, математические и естественнонаучные дисциплины" (ЕН.В.21).</p> <p>Рабочая программа обсуждена на заседании кафедры вычислительной техники. Протокол № 1 от 30 января 2006 г.</p>
Адресат курса	<p>Студенты 1-го курса, обучающиеся по специальности - Комплексное обеспечение информационной безопасности автоматизированных систем.</p>
Основная цель (цели) дисциплины	<p>Основная цель курса: освоить понятия информатики через создание программ на языке высокого уровня (Си++).</p> <p>Данная дисциплина имеет своей целью :</p> <ul style="list-style-type: none"> - Сформировать базовое представление, первичные знания, умения и навыки студентов по основам информатики, достаточные для дальнейшего продолжения образования и самообразования в области информационных технологий, вычислительной техники и программировании. - Подготовить студентов к использованию полученных навыков при изучении дальнейших дисциплин учебного плана подготовки специалистов. <p>В рамках курса "Информатика" был освоен базовый уровень программирования. В данном курсе студенты должны перейти на следующий уровень традиционного программирования. В курсе присутствуют разделы, знание которых необходимо для объектноориентированного программирования.</p>
Ядро дисциплины	<p>Основные понятия информатики, изложенные в обязательном минимуме ГОС.</p> <p>Изучение понятий информатики через углубленное изучение языка программирования СИ, СИ++.</p>

	<p>ПРЕДМЕТЫ ИЗУЧЕНИЯ</p> <ul style="list-style-type: none"> - Базовые определения и понятия информатики. - Основы программирования на языке Си++. - Программное обеспечение и технологии программирования
Связи с другими учебными дисциплинами основной образовательной программы	<p>Изучение языка происходит в несколько этапов в течение 3 семестров, на каждом из которых достигается определенный уровень качества и понимания как предмета в целом, так и навыков программирования. Курс занимает промежуточное место в триаде тесно связанных дисциплин: "Информатика" (1 семестр), "Дополнительные главы информатики" (2 семестр) и "Методы программирования" (3 семестр). Эти три дисциплины образуют по сути единый курс продолжительностью 3 семестра.</p> <p>Прохождение данного курса во втором семестре дает возможность углубленного изучения дисциплины "Языки программирования" на втором курсе.</p>
Требования к первоначальному уровню подготовки обучающихся	<p>Дисциплина базируется на школьных знаниях математики и информатики и на знаниях, полученных в результате прохождения обучения по дисциплине "Информатика" в первом семестре.</p>
Особенности организации учебного процесса по дисциплине	<p>Все понятия информатики излагаются исключительно через язык программирования, который является не столько предметом изучения, сколько инструментом решения задач. Такой инструментальный подход позволяет избежать схоластического подхода и сформировать необходимые представления и понятия отталкиваясь от практики. В результате этого курса студенты получают навыки объектно-ориентированного программирования.</p>

3. Цели учебной дисциплины

Таблица 3.1

После изучения дисциплины студент будет

иметь представление	
1	об общем понятии информации;
2	о развитии языков программирования;
3	о реализации операций и операторов языка высокого уровня на языке ассемблера;
4	о системном программном обеспечении;
знать	
5	общую характеристику процесса сбора, передачи, обработки и накопления информации;
6	структуру вычислительных машин, принципы их функционирования, основные характеристики, историю развития;
7	основные типы ЭВМ: архитектуру, назначение центрального процессора и основной памяти, классы памяти, типы и назначение внешних устройств;
8	операционные системы для различных ЭВМ: файловые системы, система управления работой пользователей, командные языки; трансляторы и редакторы связей;
9	трансляторы и редакторы связей;
уметь	
11	работать на современных ПЭВМ на уровне пользователя под управлением основных операционных систем;
12	использовать современные средства разработки программного обеспечения, включая современный пользовательский интерфейс, на языках высокого уровня, библиотеки объектов и классов для решения задач создания и сопровождения автоматизированных систем;
13	использовать технологии объектно-ориентированного программирования
15	выбирать оптимальные в конкретных условиях структуры данных и алгоритмы;
иметь опыт (владеть)	
16	современными средствами разработки и отладки программ;
17	программированием на языках Си, Си++

4. Содержание и структура учебной дисциплины

Лекционные занятия

Таблица 4.1

(Модуль), дидактическая единица, тема	Часы	Ссылки на цели
Семестр: 2		
Модуль: Основные понятия информатики		
Дидактическая единица: общая характеристика процесса сбора, передачи, обработки и накопления информации;		

Моделирование	2	1, 15, 5
Модуль: Объектно-ориентированное программирование		
Дидактическая единица: структура программы на языке высокого уровня, представление текста программы, оформление программы;		
Начальные понятия об объектно-ориентированном программировании	2	11, 12, 13, 16, 2, 4, 6, 7
Конструкторы и деструкторы	2	11, 12, 13, 2, 6
Функции друга (friend)	2	11, 12, 13, 2, 6
Перегрузка операций	4	11, 12, 13, 2, 6
Производные классы	4	11, 12, 13, 2, 6
Полиморфизм	2	11, 12, 13, 2, 6
Абстрактные классы	2	11, 12, 13, 2, 6
Потоки	2	11, 12, 13, 2, 6
Форматирование потоков	2	11, 12, 13, 2, 6
Файловый ввод-вывод с применением потоков	2	11, 12, 13, 2, 6
Шаблоны	4	11, 12, 13, 2, 6
Обработка исключений		11, 12, 13, 2, 6
Дидактическая единица: развитие языков программирования;		
Введение в C++	2	13, 16, 17, 2, 4, 6, 8, 9

Практические занятия

Таблица 4.2

(Модуль), дидактическая единица, тема	Учебная деятельность	Часы	Ссылки на цели
Семестр: 1			
Модуль: Объектно-ориентированное программирование			
Дидактическая единица: развитие языков программирования;			
Начальные понятия об объектно-ориентированном программировании	Начальные понятия об объектно-ориентированном программировании	2	12, 13, 16, 17, 2
Конструкторы и деструкторы	Конструкторы и деструкторы	4	12, 13, 16, 17, 2, 7
Функции друга	Функции друга	2	11, 12, 13, 16, 17, 2, 9
Перегрузка операций	Перегрузка операций	2	12, 13, 16, 17, 2
Производные классы	Производные классы	2	12, 13, 16, 17, 2
Полиморфизм	Полиморфизм	4	12, 13, 16, 17, 2
Абстрактные классы	Абстрактные классы	2	12, 13, 16, 17, 2

Модуль: Технология программирования			
Дидактическая единица: развитие языков программирования;			
Шаблоны	Шаблоны	4	12, 13, 16, 17, 2
Обработка исключений	Обработка исключений	4	12, 13, 16, 17, 2
Потоки	Программирование потоков. Форматирование потоков.	4	12, 13, 16, 17, 2
Обработка исключений	Обработка исключений	2	12, 13, 16, 17, 2

Лабораторная работа

Таблица 4.3

(Модуль), дидактическая единица, тема	Учебная деятельность	Часы	Ссылки на цели
Семестр: 1			
Модуль: Объектно-ориентированное программирование			
Дидактическая единица: развитие языков программирования;			
Начальные понятие об объектно-ориентированном программировании	Общее понятие классов. Синтаксис и семантика классов на C++.	4	12, 13, 17, 2
Конструкторы и деструкторы	Конструкторы и деструкторы	4	12, 16, 17, 2
Функции друзья	Функции друзья	4	12, 13, 16, 17, 2
Перегрузка операций	Перегрузка операций	4	12, 13, 16, 17, 2
Производные классы	Производные классы	4	12, 13, 16, 17, 2
Полиморфизм	Полиморфизм	4	12, 13, 16, 17, 2
Абстрактные классы	Абстрактные классы	4	12, 13, 16, 17, 2
Модуль: Технология программирования			
Дидактическая единица: развитие языков программирования;			
Шаблоны	Шаблоны	4	12, 13, 16, 17, 2
Обработка исключений	Обработка исключений	4	12, 13, 16, 17, 2

5. Самостоятельная работа студентов

Семестр- 2, Подготовка к зачету

Для подговки к зачету необходимо 6 часов самостоятельной работы.

Семестр- 2, Подготовка к занятиям

Для подготовки к практике необходимо 3 часа.

8 практических заданий - 24 часа

6. Правила аттестации студентов по учебной дисциплине

Для аттестации студентов по дисциплине используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате итоговой аттестации (экзамен, защита курсовой работы).

В таблице 1 приведено максимальное количество баллов, которое может набрать студент по видам учебной деятельности в течение семестра и диапазоны баллов, соответствующие минимальному и максимальному количеству баллов. Максимальная сумма баллов за семестр составляет 100 баллов (текущий рейтинг - 60 баллов, итоговая аттестация - 40 баллов).

Правила текущей аттестации в 2-м семестре:

1. В течение второго семестра необходимо выполнить и защитить 8 лабораторных работ в сроки, установленные учебным графиком (см. таблицу 1).

2. К защите допускаются студенты, выполнившие задание на лабораторную работу и оформившие отчет по работе в соответствии с обычными требованиями.

3. На защите предлагается два теоретических вопроса и один практический вопрос (по ходу выполнения работы).

4. Максимальное количество баллов (указанное в графе 3 таблицы 1) выставляется, если студент полностью ответил на все вопросы, без серьезных замечаний и недочетов.

5. Среднее количество баллов (согласно графе 4 таблицы 1) выставляется, если студент полностью ответил на два вопроса из трех, причем один из этих вопросов - практический.

6. Минимальное количество баллов (согласно графе 4 таблицы 1) выставляется, если студент ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.

7. Передача лабораторной работы назначается, если студент не ориентируется в учебном материале, не может объяснить ход и результаты выполнения работы. В случае передачи работы максимально может быть выставлено не более, чем среднее от указанного в графе 4 таблицы 1 количества баллов.

8. В случае представления и защиты работ с опозданием от учебного графика количество баллов уменьшается (опоздание на 1 неделю - потеря 1 балла, опоздание на 2 недели - потеря 2 баллов, 3 недели и более - потеря 50% баллов от максимально возможного).

Таблица 1.

№	Вид учебной работы выполнения/защиты	Макс. баллов	Диапазон баллов	Неделя
1	2	3	4	5
1	Лабораторная работа №1	8	4-8	3
/ 5				
2	Лабораторная работа №2	8	4-8	5
/ 7				
3	Лабораторная работа №3	8	4-8	7
/ 9				
4	Лабораторная работа №4	8	4-8	9
/ 11				
5	Лабораторная работа №5	8		4-8
11 / 13				

6	Лабораторная работа №6	8	4-8
13 / 15			
7	Лабораторная работа №7	8	4-8
15 / 16			
8	Лабораторная работа №8	4	2-4
17 / 17			
	Итого по текущему рейтингу	60	30-60

Правила итоговой аттестации в 2-м семестре:

1. К экзамену допускаются студенты, сдавшие и защитившие все лабораторные работы и набравшие не менее 50% (30 баллов) по результатам текущего рейтинга.

2. Экзамен проводится в устном виде, предлагается один теоретический вопрос и одно практическое задание (общий перечень экзаменационных билетов прилагается).

3. Максимальное количество 36-40 баллов выставляется, если ответ на теоретический вопрос полностью и по существу раскрывает суть вопроса и правильно выполнено практическое задание.

4. Количество баллов 28-35 выставляется, если по теоретическому вопросу дан неполный ответ или практическое задание выполнено с незначительными ошибками.

5. Количество баллов 20-29 выставляется, если ответы на оба вопроса содержали существенные ошибки.

Сумма баллов текущего рейтинга и баллов, полученных за экзамен, является итоговым рейтингом за 1-й семестр.

Формирование оценки по набранным баллам в 2-м семестре:

85 -100	5
68 - 84	4
50 - 67	3

Формирование оценки ECTS в 1-м семестре:

97 - 100	A+
93 - 96	A
89 - 92	A-
85 - 88	B+
81 - 84	B
77 - 80	B-
63 - 76	C+
72 - 75	C
68 - 71	C-
65 - 67	D+
62 - 64	D
59 - 61	D-
56 - 58	E+
53 - 55	E
50 - 52	E-
Меньше 50	F

7. Список литературы

7.1 Основная литература

В печатном виде

1. Керниган Б. В. Язык программирования Си / Б. Керниган, Д. Ритчи ; пер. с англ. Вс. С. Штаркмана. - СПб., 2004. - 351 с.
2. Подбельский В. В. Язык Си++ : учебное пособие для вузов по направлениям "Прикладная математика" и "Вычислительные машины, комплексы, системы и сети" / В. В. Подбельский. - М., 2006. - 559 с. : ил., табл. - Рекомендовано МО.
3. Подбельский В. В. Программирование на языке Си : учебное пособие для вузов по направлениям: "Прикладная математика и информатика", "Информатика и вычислительная техника", специальностям "Прикладная математика", "Вычислительные машины, комплексы, системы и сети управления" / В. В. Подбельский, С. С. Фомин. - М., 2007. - 600 с. : ил., табл. - Рекомендовано МО.
4. Подбельский В. В. Практикум по программированию на языке Си : учебное пособие для вузов по направлениям "Прикладная математика и информатика", "Информатика и вычислительная техника" и специальности "Прикладная математика и информатика" / В. В. Подбельский. - М., 2004. - 574, [1] с. + 1 электрон. опт. диск (CD-ROM). - Рекомендовано МО.
5. Фридман А. Л. Язык программирования Си++. Курс лекций : учебное пособие / А. Л. Фридман ; Интернет ун-т информ. технологий. - М., 2004. - 257 с. : ил., схемы - Рекомендовано УМО.
6. Романов Е. Л. Практикум по программированию на С++ : [учебное пособие] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - СПб., 2004. - 426, [1] с. : ил.

В электронном виде

1. Романов Е. Л. Практикум по программированию на С++ : [учебное пособие] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - СПб., 2004. - 426, [1] с. : ил.

7.2 Дополнительная литература

В печатном виде

1. Романов Е. Л. Язык Си++ в задачах, вопросах и ответах : [учебное пособие] / Е. Л. Романов. - Новосибирск, 2003. - 426, [1] с. : ил.

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

В печатном виде

1. Разработка простых консольных приложений с помощью Microsoft Visual Studio 2010 : методические указания к лабораторным работам для 1 курса ФПМИ (направление 010500 - Прикладная математика и информатика, специальность 010503 - Математическое и программное обеспечение и администрирование информационных систем) дневного отделения / Новосиб. гос. техн. ун-т ; [сост.: Ю. В. Тракимус, А. Г. Задорожный, А. В. Чернышев]. - Новосибирск, 2011. - 41, [3] с. : ил., табл.

В электронном виде

1. Юн С. Г. Технология разработки программ и ПО [Электронный ресурс] : электронный учебно-методический комплекс / С. Г. Юн ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=74>. - Загл. с экрана.
2. Коротиков С. В. Технология программирования и разработки программного обеспечения [Электронный ресурс] : электронный учебно-методический комплекс / С. В. Коротиков ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=1255>. - Загл. с экрана.
3. Разработка простых консольных приложений с помощью Microsoft Visual Studio 2010 : методические указания к лабораторным работам для 1 курса ФПМИ (направление 010500 - Прикладная математика и информатика, специальность 010503 - Математическое и программное обеспечение и администрирование информационных систем) дневного отделения / Новосиб. гос. техн. ун-т ; [сост.: Ю. В. Тракимус, А. Г. Задорожный, А. В. Чернышев]. - Новосибирск, 2011. - 41, [3] с. : ил., табл.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2011/11_4009.pdf
4. Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=351>. - Загл. с экрана.

8.2 Программное обеспечение

1. Microsoft, Visual Studio, Интегрированная среда разработки

9. Контролирующие материалы для аттестации студентов по дисциплине

Билет 1

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.
 2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `int f(char *a, char * b);`
Параметры: a - указатель исходной строки; b - указатель результирующей строки
Назначение: отсортировать символы в строке в алфавитном порядке. Алгоритм: поиск минимального в исходной строке, запись в выходную строку.
-

Билет 2

1. Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.
 2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `void f(char *a);`
Параметры: a - указатель исходной строки
Назначение: Переставить в строке символы попарно слева и справа (1..n, 2..n-1 и т.д.);
-

Билет 3

1. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные). Приоритеты и порядок выполнения.
 2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `void f(char *a);`
Параметры: a - указатель исходной строки
Назначение: Отсортировать символы в строке. Алгоритм: последовательно сравниваются пары соседних элементов x_k и x_{k+1} ($k=0, \dots, n-2$), если $x_k > x_{k+1}$, то они переставляются.
-

Билет 4

1. Операторы языка. Составной оператор. Управляющие конструкции языка.
 2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `void f(char *a);`
Параметры: a - указатель исходной строки
Назначение: Вывести на экран в первой строке нечетные символы, во второй четные символы исходной строки.
-

Билет 5

1. Условный оператор (if ... else ...). Переключатель (switch).
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main().
Прототип: `void f(char *a);`
Параметры: a - указатель исходной строки

Назначение: Вывести на экран сначала цифры, затем буквы, затем остальные символы исходной строки.

Билет 6

1. Операторы цикла (while , do while, for)
2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main().

Прототип: void f(char *a);

Параметры: a - указатель исходной строки

Назначение: Для каждого символа исходной строки найти число вхождений его в строке и напечатать в столбец (символ, частота вхождения).

Билет 7

1. Операторы передачи управления (break , continue , goto , return)
2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f(char *a , char * b);

Параметры: a - указатель исходной строки

b - указатель результирующей строки

Назначение: Исключить из исходной строки все пробелы и записать в результирующую строку

Билет 8

1. Функции в С. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *mat, int n, int m, int &min, int &max);

Параметры: mat - указатель исходной матрицы

n - количество строк

m - количество столбцов

min - минимальное значение

max - максимальное значение

Назначение: Вычисление максимального и минимального значений элементов матрицы

Билет 9

1. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *mat, int n, int m, int & A, int & B);

Параметры: mat - указатель исходной матрицы

n - количество строк

m - количество столбцов

A - число элементов > 0

B- число элементов < 0

Назначение: Вычисление числа элементов больше нуля и меньше нуля

Билет 10

1. Указатели на массивы. Массивы указателей и многомерные массивы. Динамическое выделение памяти под массивы. Инициализация указателей

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *mat, int n);

Параметры: mat - указатель квадратной матрицы

n - количество строк, столбцов

m - количество столбцов

Назначение: Заполнение матрицы последовательностью чисел от 0 до n*n, расположенными по спирали

```
1  2  3  4  5  6
20 21 22 23 24  7
19 32 33 34 25  8
18 31 36 35 26  9
17 30 29 28 27 10
16 15 14 13 12 11
```

Билет 11

1. Общая структура программы. Время существования и видимость переменных. Классы памяти

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f(int *A, int *B, int n, int m);

Параметры: A - указатель исходной матрицы

B - указатель транспонированной матрицы

n - количество строк исходной матрицы

m - количество столбцов исходной матрицы

Назначение: Нахождение транспонированной матрицы.

Транспонирование заключается в обмене местами строк и столбцов матрицы. Элемент, расположенный в i-той строке и j-том столбце помещается в j-тую строку и i-тый столбец.

Билет 12

1. Структуры, объединения, перечисления

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f(int *A, int *X, int *Y, int n, int m);

Параметры: A - указатель матрицы

X - указатель вектора (одномерного массива)

Y - указатель результирующего вектора

n - количество строк матрицы

m - количество столбцов матрицы

Назначение: вычисление произведения матрицы на вектор.

В результате умножения матрицы A[m][n] на вектор X[n] получается вектор Y[m],

компоненты которого вычисляются по формуле: $y[i]=\text{SUM}(a[i][j]*x[j])$, (i = 0, 2, ..., m-1)

Билет 14

1. Рекурсивные вызовы функций. Аргументы командной строки.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f(int *A, int *B, int *C, int n, int m, int l);

Параметры: A - указатель первой матрицы

B - указатель второй матрицы

C - результирующей матрицы

n - количество строк первой матрицы

m - количество столбцов первой матрицы, количество строк второй матрицы

l - количество столбцов второй матрицы

Назначение: вычисление произведения 2-х матриц (строка на столбец).

В результате умножения двух матриц A[m][n] и B[n][l] получается матрица C[m][l], элементы которой находятся по следующей формуле: $C[i][j] = \text{SUM } a[i][k] * b[k][j]$

Билет 15

1. Внешние объекты программы (extern). Статические переменные и функции (static). Регистровые переменные (register).
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *A, int m);

Параметры: A - указатель квадратной матрицы

m - количество строк, столбцов матрицы

Назначение: вычисление среднего арифметического элементов, лежащих ниже диагонали

Билет 16

1. Препроцессор языка Си. (#define #undef #include #if, #ifdef, #ifndef, #else, #elif и #endif)
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *a, int n);

Параметры: a - указатель исходного массива, n - размер исходного массива.

Назначение: в исходном массиве все элементы, встречающиеся больше одного раза заменить на нули.

Билет 17

1. Общая структура программы. Время существования и видимость переменных. Классы памяти
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int *a, int n);

Параметры: a - указатель исходного массива, n - размер исходного массива.

Назначение: Отсортировать элементы массива по убыванию методом пузырька

Алгоритм : последовательно сравниваются пары соседних элементов x_k и x_{k+1} ($k=0, n-2$),

если $x_k < x_{k+1}$, то они переставляются.

Билет 18

1. Указатели на массивы. Массивы указателей и многомерные массивы. Динамическое выделение памяти под массивы. Инициализация указателей
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `int f(int *a, int n);`
Параметры: a - указатель исходного массива, n - размер исходного массива.
Назначение: Отсортировать элементы массива по следующему алгоритму: отыскивается максимальный элемент и переносится в конец массива, затем метод переносится ко всем элементам, кроме последнего и т.д.

Билет 19

1. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.
 2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
Прототип: `int f(int M, int N);`
Параметры: M,N - целые числа
Назначение: Нахождение наибольшего общего делителя двух чисел N и M по рекуррентному соотношению
$$N_k = N_{k-2} - \text{INT}(N_{k-2} / N_{k-1}) \quad N_{k-1} \quad k=2,3 \dots$$
$$N_0 = \max(|N|, |M|) \quad N_1 = \min(|N|, |M|)$$
Если $N_k = 0 \Rightarrow \text{НОД} = N_{k-1}$
(N=23345 M=9135 => 1015 N=238 M=347 => 34)
-
-

Билет 20

1. Функции в C. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
 - 2.1. Прототип: `int f(int *a, int n);`
 - 2.2. Параметры: a - указатель исходного массива, n - размер исходного массива.
 - 2.3. Назначение: в исходном массиве удаляются все элементы, значения которых меньше, чем количество их повторов в массиве. Порядок следования остальных элементов массива не изменяется.
 - 2.4. Возвращаемое значение: количество оставшихся элементов.
 - 2.5. Пример использования:

```
int a[] = { 1, 2, 3, 4, 2, 4, 6, 2, 4, 7, 1, 1, 5 };
int n;
printf( "n = %d\n", n = f( a, sizeof(a)/sizeof(int)));
for( int i = 0 ; i < n ; printf( "%d ", a[i++]));
// вывод на экран:
//
// n = 7
// 3 4 4 6 4 7 5
```

Билет 21

1. Операторы передачи управления (break , continue , goto , return)

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
 - 2.1. Прототип: int f(int* a, int n, int k);
 - 2.2. Параметры: a - указатель исходного массива, n - размер массива, k - размер сдвига.
 - 2.3. Назначение: исходный массив преобразуется в результате циклического сдвига на указанное число позиций. "Циклически" означает, что при выходе за верхнюю границу массива запись производится в позиции 0, 1 и т.д., а при выходе за нижнюю - в позиции n-1, n-2 и т.д.
 - 2.4. Возвращаемое значение: свой первый параметр.
 - 2.5. Пример использования:


```
int i, x[] = { 3, 6, 4, 1, 8, 2, 5, 7, 0 };
f( x, sizeof(x)/sizeof(int), -3 );
for( i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
f( x, sizeof(x)/sizeof(int), 2 ); printf( "\n" );
for( i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
// вывод на экран:
//
// 1 8 2 5 7 0 3 6 4
// 6 4 1 8 2 5 7 0 3
//
```

Билет 22

1. Условный оператор (if ... else ...). Переключатель (switch).
2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
 - 2.1. Прототип: int f(char *a, float *b);
 - 2.2. Параметры: a - указатель исходной строки, b - указатель буфера.
 - 2.3. Назначение: в исходной строке ищутся и распечатываются в столбик все последовательности литер, представляющие записи целых чисел или действительных чисел в формате <целая часть>.<дробная часть>. Сами числа в формате float последовательно записываются в буфер.
 - 2.4. Возвращаемое значение: количество найденных и записанных чисел.
 - 2.5. Пример использования:


```
char s[] = "12.45, 678.99, -34 0.0004 ";
float B[100];
printf( "\nЗаписано чисел: %d", f( s, B ) );
// вывод на экран:
//
// 12.450000
// 678.990000
// -34.000000
// 0.000400
// Записано чисел: 4
//
```

Билет 23

1. Операторы языка. Составной оператор. Управляющие конструкции языка.
2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
 - 2.1. Прототип: int* f(int* a, int n);
 - 2.2. Параметры: a - указатель исходного массива, n - размер массива.

2.3. Назначение: исходный массив преобразуется по следующему алгоритму: все нечетные элементы массива группируются в порядке возрастания в начале массива, а четные в порядке убывания в конце.

2.4. Возвращаемое значение: указатель первого четного элемента в преобразованном массиве.

2.5. Пример использования:

```
int x[] = { 3, 6, 4, 8, 2, 5, 7, 0 };
int *p = f( x, sizeof(x)/sizeof(int));
printf( "n = %d\n", p-x );
for( int i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
// вывод на экран:
//
// n = 3
// 3 5 7 8 6 4 2 0
//
```

Билет 24

1. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные). Приоритеты и порядок выполнения.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

2.1. Прототип: char* f(char *v1, char *v2, char *r, int n);

2.2. Параметры: v1, v2 - указатели исходных строк, r - указатель результирующей строки, n - размер результирующей строки.

2.3. Назначение: исходные строки являются символьным представлением натуральных чисел произвольной длины. Необходимо получить в результирующей строке такое же представление суммы этих чисел. Не уместающиеся в результирующей строке цифры пропадают. Остающиеся свободные позиции заполняются кодом нуля.

2.4. Возвращаемое значение: указатель литеры, являющейся представлением первой значащей цифры результата.

2.5. Пример использования:

```
char a[] = "45789", b[] = "1111111", c[20];
printf( "\n%s\n%s", f( a, b, c, sizeof c ), c );
//
// вывод на экран:
//
// 1156900
// 000000000000001156900
//
```

Билет 25

1. Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(int n);

Назначение: $y = 1! + 2! + 3! + 4! + \dots + n!$

Билет 26

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$e^x = 1 + x + x^2/2! + x^3/3! + \dots$$

Билет 27

1. Препроцессор языка Си. (#define #undef #include #if, #ifdef, #ifndef, #else, #elif и #endif)

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\sin(x) = x - x^3/3! + x^5/5! + \dots$$

Билет 28

1. Внешние объекты программы (extern). Статические переменные и функции (static). Регистровые переменные (register).

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\cos(x) = 1 - x^2/2! + x^4/4! \dots$$

Билет 29

1. Рекурсивные вызовы функций. Аргументы командной строки.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\ln(1+x) = x - x^2/2 + x^3/3 - \dots \quad |x| < 1$$

Билет 30

1. Структуры, объединения, перечисления

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\cos(x) = 1 - x^2/2! + x^4/4! \dots$$

Билет 31

1. Общая структура программы. Время существования и видимость переменных. Классы памяти

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\text{tg}(x) = \frac{x}{1 - \frac{x^2}{3 - \frac{x^2}{5 - \frac{x^2}{7 - \dots}}}}$$

Билет 32

1. Указатели на массивы. Массивы указателей и многомерные массивы. Динамическое выделение памяти под массивы. Инициализация указателей

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$\text{sqrt}(2) = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}}$$

Билет 33

1. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

$$e = 2 + \frac{2}{3} + \frac{2}{4} + \frac{2}{5} + \dots$$

Билет 34

1. Функции в С. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: double f(double x, double e);

Найти функцию с точностью ϵ , т.е. пока $|f_{n+1}-f_n| < \epsilon$. Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.

Билет 35

1. Операторы передачи управления (break , continue , goto , return)

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f(int n);

Печать римских цифр

- | | | |
|----|-------|--------|
| 1. | 1- I | 10-X |
| 2. | 2-II | 50-L |
| 3. | 3-III | 100-C |
| 4. | 4-IV | 500-D |
| 5. | 5-V | 1000-M |
| 6. | 6-VI | |

Соблюдая следующие правила:

7. Если младшая цифра стоит перед старшей она вычитается IV-4

8. Если младшая цифра стоит после старшей - складывается VI -5

9. Число представляется минимальным количеством знаков III-3, 4-IV, а не IIII.

10.

Билет 36

1. Операторы цикла (while , do while, for)

2. Написать на языке "С" или "С++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: int f(int n, int k);

Напечатать подряд все натуральные числа до n - 012345678910111213 Найти k-ую цифру последовательности

Билет 37

1. Условный оператор (if ... else ...). Переключатель (switch).
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям

Найти все четырехзначные числа Армстронга. Натуральное число из n знаков является числом Армстронга, если сумма его цифр, возведенная в n-ю степень равна самому числу.
 $153=1^3+5^3+3^3$

Билет 38

1. Операторы языка. Составной оператор. Управляющие конструкции языка.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям

Вставить между цифрами 1,2,3,4...9 знак одной из операций +, -, *, /, так, чтобы результат 8-опреаций равнялся 100. Имеется 8 решений. Распечатать их со знаками.
 $1*2+3*4*5-6+7+8-9=100$

Билет 39

1. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные). Приоритеты и порядок выполнения.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям

Совершенными числами называются числа, равные сумме своих делителей ($6=1+2+3$). Найти все совершенные числа от 0 до 10000 и распечатать их вместе со своими делителями.

Билет 40

1. Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f1(int n, *char a); // a - массив, состоящий из 0 или 1. В main() массив распечатать

Перевести из десятичной системы в двоичную целое число

$327 : 2 = 163$ остаток 1

$163 : 2 = 81$ 1

$81 : 2 = 40$ 1

$40 : 2 = 20$ 0

$20 : 2 = 10$ 0

$10 : 2 = 5$ 0

$5 : 2 = 2$ 1

$2 : 2 = 1$ 0

$1 : 2 = 0$ 1

$32710 = 1010001112$

Билет 41

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.

2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()

Прототип: void f1(double b, *char a); // a - массив, состоящий из 0 или 1. В main() массив распечатать

Перевести из десятичной системы в двоичную дробное

$$0.125 \times 2 = 0.250 = 0 + 0.250$$

$$0.250 \times 2 = 0.5 = 0 + 0.5$$

$$0.500 \times 2 = 1.000 = 1 + 0.00$$

$$0.12510 = 0.0012$$