

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Языки программирования

: 12.03.04

, :

: 1, : 2

,

		2
1	()	4
2		144
3	, .	64
4	, .	18
5	, .	0
6	, .	36
7	, .	18
8	, .	2
9	, .	8
10	, .	80
11	(, ,)	
12		

(): 12.03.04

216 12.03.2015 ., : 08.04.2015 .

: 1,

(): 12.03.04

, 2/1 20.06.2017

, 6 21.06.2017

:

,

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.9 способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности; в части следующих результатов обучения:				
1.				
1.			-	
6.				

2.

2.1

	(
--	---	--	--	--	--

.9. 1				
1.знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных				
.9. 1				
2.знать методы построения современных проблемно-ориентированных прикладных программных средств				
.9. 6				
3.иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения				

3.

3.1

: 2				
1.	0	1	1	
2.				
(0	1	1	

3. ++	0	1	1, 2, 3	
4. -	0	1	1, 2, 3	
5.	0	1	1, 2, 3	
5.	0	1	1, 2, 3	
6.	0	1	1, 2, 3	
7.	0	1	2, 3	
8.	0	2	1, 2, 3	
9.	0	2	1, 2, 3	
10. . -	0	2	1, 2, 3	
11.	0	2	1, 2, 3	
12.	0	2	1, 2, 3	

3.2

	,	.		
: 2				
: ++				
1. ++	2	4	1, 2, 3	
2. .	2	4	1, 2, 3	
3.	2	4	1, 2, 3	
4.	2	4	1, 2, 3	
5.	2	4	1, 2, 3	
6.	2	4	1, 2, 3	
7.	2	4	1, 2, 3	
8.	2	4	1, 2, 3	
9.	2	4	1, 2, 3	

4.

: 2				

1		1, 2, 3	40	4
[]: - / . . . ; , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000157445. -				
2		1, 2, 3	20	0
: []: - / . . . ; , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000157445. -				
3		1, 2, 3	20	4
: []: - / . . . ; , [2011]. - : http://elibrary.nstu.ru/source?bib_id=vtls000157445. -				

5.

, (. 5.1).

5.1

	-
	e-mail

5.2

1	
Краткое описание применения:	

6.

(),

15-

ECTS.

. 6.1.

6.1

	.	
: 2		
Лекция:	10	10
Лабораторная:	13	25
РГЗ:	12	25

Экзамен:	20	40
----------	----	----

6.2

6.2

[illegible]

1

7.

1. Прата С. Язык программирования C++ : лекции и упражнения / Стивен Прата ; [пер. с англ. Д. Я. Иваненко и др.]. - М. [и др.], 2007. - 1181 с. : ил.
 2. Павловская Т. А. C/C++. Структурное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - СПб. [и др.], 2007. - 238 с. : ил.. - На тит. л.: Издательская программа 300 лучших учебников для высшей школы.
 3. Глушаков С. В. Программирование на C++ / С. В. Глушаков, Т. В. Дуравкина. - Москва, 2008. - 685, [1] с.
-
1. Савитч У. Язык C++ . Курс объектно-ориентированного программирования / Уолтер Савитч ; [пер. с англ. И. В. Красикова, В. Н. Романова, А. В. Скилягина]. - Москва, 2001. - 696 с. : ил.. - Парал. тит. л. англ..
 2. Пол И. Объектно-ориентированное программирование с использованием C++ : Пер. с англ.. - Киев, 1995. - 480 с.
 3. Дьюхарст С. Программирование на C++ : Пер. с англ. / С. Дьюхарст. - Киев, 1993. - 272 с. : ил.
 4. Савич У. Программирование на C++ / У. Савич ; [пер. с англ. О. Здир]. - Санкт-Петербург [и др.], 2004. - 780 с. : ил.. - Парал. тит. л. англ..
 5. Тяпичев Г. А. Быстрое программирование на C++ / Г. А. Тяпичев. - Москва, 2004. - 372 с. : ил., табл.. - Компакт-диск отсутствует.
 6. Кьюу Д. Объектно-ориентированное программирование : учебный курс / Дж. Кьюу, М. Джеанини. - СПб., 2005. - 237 с. : ил.
 7. Фленов М. Е. Программирование на C++ глазами хакера / Михаил Фленов. - СПб., 2009. - 350 с. : ил. + 1 CD-ROM.
 8. C++ язык программирования. - М., 1991. - 315 с.
 9. Двоеглазов И. М. Язык программирования C++ : справочное руководство / И. М. Двоеглазов. - Киев, 1993. - 128 с.

10. Страуструп Б. Язык программирования СИ++ / Пер. с англ. Пиголкина М. Г. , Яницкого В. А. - М., 1991. - 348 с. : ил.

11. Прата С. Язык программирования С++. Лекции и упражнения : учебник : [пер. с англ.] / Стивен Прата. - Санкт-Петербург [и др.], 2005. - 1097 с. : ил.. - Парал. тит. л. англ..

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>

2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>

3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>

4. ЭБС "Znanium.com" : <http://znanium.com/>

5. :

8.

8.1

1. Лисицин Д. В. Программирование. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000157445. - Загл. с экрана.

8.2

1 Visual Studio 2010

9.

1	(- , ,)	

1	(Internet)	

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра систем сбора и обработки данных

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н. Рева И. Л.

“ ” _____ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Языки программирования

Образовательная программа: 12.03.04 Биотехнические системы и технологии

Факультет автоматики и вычислительной техники

Обобщенная структура фонда оценочных средств учебной дисциплины

Тема	Код формируемой компетенции	Знания/умения	Контролирующее мероприятие (экзамен, зачет, курсовой проект и т.п.)
Классификация языков программирования (Процедурные, Функциональные, Логические, Объектно-ориентированные)	ОПК.9	з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных	Лабораторная Экзамен
История развития языков программирования		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных	Лабораторная Экзамен
Программирование потоков		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств	Лабораторная РГЗ Экзамен
Программирование классов. Конструкторы и деструкторы.		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств	Лабораторная РГЗ Экзамен
Отличие C++ от C		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств	Лабораторная Экзамен
Перегрузка операций		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств у6. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная РГЗ Экзамен
Полиморфизм		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств у6. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен

Потоки. Форматирование потоков. Файловый ввод-вывод с применением потоков.	ОПК.9	з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная РГЗ Экзамен
Производные классы		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен
C++ как улучшенный C		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен
Объектно-ориентированное программирование		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен
Конструкторы и деструкторы		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная РГЗ Экзамен
Исключения		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен

Шаблоны	ОПК.9	з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен
Абстрактные классы		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная Экзамен
Друзья		з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных у1. знать методы построения современных проблемно-ориентированных прикладных программных средств уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Лабораторная РГЗ Экзамен

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 (посещение лекций, выполнение лабораторных работ) и баллов, полученных в результате итоговой аттестации (экзамен, РГЗ).

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

Правила текущей аттестации в 3-м семестре:

1. В течение второго семестра необходимо выполнить и защитить 8 лабораторных работ в сроки, установленные учебным графиком .
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 выставляется, если студент полностью ответил на все вопросы, без серьезных замечаний и недочетов.

5. Среднее количество баллов выставляется, если студент полностью ответил на два вопроса из трех, причем один из этих вопросов - практический.
6. Минимальное количество баллов выставляется, если студент ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.
7. Пересдача лабораторной работы назначается, если студент не ориентируется в учебном материале, не может объяснить ход и результаты выполнения работы.
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).

Правила итоговой аттестации:

1. К экзамену допускаются студенты, сдавшие и защитившие все лабораторные работы и набравшие не менее 50% по результатам текущего рейтинга.
 2. Экзамен проводится в устном виде, предлагается один теоретический вопрос и одно практическое задание.
 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.

Примерный список вопросов и практических заданий приводится в Приложении

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине программирования приведена в Таблице.

Языки

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.9 способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности	з1. знать технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных	Абстрактные классы Друзья Исключения История развития языков программирования Классификация языков программирования (Процедурные, Функциональные, Логические, Объектно-ориентированные) Конструкторы и декструкторы Объектно-ориентированное программирование Отличие С++ от С Перегрузка операций Полиморфизм Потоки. Форматирование потоков. Файловый ввод-вывод с применением потоков. Программирование классов. Конструкторы и декструкторы. Программирование потоков Производные классы С++ как улучшенный С Шаблоны	Отчет по лабораторной работе РГЗ,	Экзамен
ОПК.9	у1. знать методы построения современных проблемно-ориентированных прикладных программных средств	Абстрактные классы Друзья Исключения Конструкторы и декструкторы Объектно-ориентированное программирование Отличие С++ от С Перегрузка операций Полиморфизм Потоки. Форматирование потоков. Файловый ввод-вывод с применением потоков. Программирование классов. Конструкторы и декструкторы. Программирование потоков Производные классы С++ как улучшенный С Шаблоны	Отчет по лабораторной работе РГЗ	Экзамен
ОПК.9	уб. иметь навык работы с типовыми пакетами прикладных программ, применяемых при проектировании аппаратов, приборов и систем медицинского назначения	Абстрактные классы Друзья Исключения Конструкторы и декструкторы Объектно-ориентированное программирование Перегрузка операций Полиморфизм Потоки. Форматирование потоков. Файловый ввод-вывод с применением потоков. Производные классы С++ как улучшенный С Шаблоны	Отчет по лабораторной работе РГЗ	Экзамен

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по **дисциплине** проводится в 2 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.9.

Кроме того, сформированность компетенции проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 2 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенции ОПК.9, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Билет 1

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.
2. Аргументы, используемые по умолчанию в C++
3. 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. Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки
3. 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. Встроенные функции
3. 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. Операция ::, перегруженные функции
3. 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. Определение переменных, константные значения в C++ (отличие от C)
3. 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. Объектно-ориентированное программирование (Инкапсуляция, наследование, полиморфизм). Понятие класса. Отличие от структур в Си.
3. 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: 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. Имена-этикетки в enum,struct и union , анонимные объединения в C++
3. Написать на языке "C" или "C++" 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. Операторы распределения памяти в С++
3. 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. Конструкторы и деструкторы
3. 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. Функции друзья
3. Написать на языке "С" или "С++" 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. Перегрузка операций
3. 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 шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки
3. 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. Производные классы. Множественное наследование
3. 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. Полиморфизм. Виртуальные функции члены. Абстрактные классы
3. 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 #endif)
2. Потоки (ввод-вывод в C++). Предопределенные объекты-потоки. Форматирование потоков. (функции-элементы, флаги, манипуляторы)
3. 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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. Обработка исключений
3. 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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

$N_k = N_{k-2} - \text{INT}(N_{k-2} / N_{k-1}) N_{k-1} \quad k=2,3 \dots$

$N_0 = \max(|N|, |M|) \quad N_1 = \min(|N|, |M|)$

Если $N_k = 0 \Rightarrow \text{НОД} = N_{k-1}$

(N=23345 M=9135 \Rightarrow 1015 N=238 M=347 \Rightarrow 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. Перегрузка операций
3. 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. Аргументы, используемые по умолчанию в C++
3. Написать на языке "C" или "C++" 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. Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки
3. Написать на языке "C" или "C++" 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
```

```
// 678.99000
```

```
// -34.000000
```

```
// 0.000400
```

```
// Записано чисел: 4
```

```
//
```


Билет 23

1. Операторы языка. Составной оператор. Управляющие конструкции языка.
2. Встроенные функции
3. 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 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. Операция ::, перегруженные функции
3. 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. Определение переменных, константные значения в C++ (отличие от C)
3. 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. Объектно-ориентированное программирование (Инкапсуляция, наследование, полиморфизм). Понятие класса. Отличие от структур в Си
3. 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` `#endif`)
2. Имена-этикетки в enum, struct и union, анонимные объединения в C++
3. 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. Операторы распределения памяти в C++
3. 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. Конструкторы и деструкторы
3. 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. Функции друзья
3. 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. Перегрузка операций
3. 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.

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

Билет 32

1. Указатели на массивы. Массивы указателей и многомерные массивы. Динамическое выделение памяти под массивы. Инициализация указателей
2. Написать шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки
3. 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.

$$\begin{array}{c} 1 \\ \text{sqrt}(2) = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}} \end{array}$$

Билет 33

1. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.
2. Написать шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки
3. 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.

$$\begin{aligned} & 2 \\ e = & 2 + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \dots \end{aligned}$$

Билет 34

1. Функции в C. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.
2. Производные классы. Множественное наследование
3. 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.

Билет 35

1. Операторы передачи управления (break , continue , goto , return)
2. Полиморфизм. Виртуальные функции члены. Абстрактные классы
3. Написать на языке "C" или "C++" 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. Поток (ввод-вывод в C++). Предопределенные объекты-поток. Форматирование потоков. (функции-элементы, флаги, манипуляторы)
3. Написать на языке "C" или "C++" 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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. Обработка исключений
3. 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-операций равнялся 100. Имеется 8 решений. Распечатать их со знаками.

$$1*2+3*4*5-6+7+8-9=10$$

Билет 39

1. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные). Приоритеты и порядок выполнения.
2. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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. Перегрузка операций
3. 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

$$327_{10} = 101000111_2$$

Билет 41

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.
2. Аргументы, используемые по умолчанию в C++
3. 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

$$\begin{array}{l} 0.125 \times 2 = 0.250 = 0 + 0.250 \\ 0.250 \times 2 = 0.5 = 0 + 0.5 \\ 0.500 \times 2 = 1.000 = 1 + 0.00 \end{array}$$

$$0.125_{10} = 0.001_2$$

Билет 42

1. Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.
2. Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки
3. 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 цифр от 0 до 7. В main() массив распечатать

Перевести из десятичной системы в восьмеричную целое число

$$\begin{array}{l} 327 : 8 = 40 \text{ остаток } 7 \\ 40 : 8 = 5 \quad \quad \quad 0 \\ 5 : 8 = 0 \quad \quad \quad 5 \end{array}$$

$$327_{10} = 0507_8$$

Билет 43

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

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

Перевести из десятичной системы в шестнадцатеричную целое число

327	:	16	=	20	остаток	7
20	:	16	=	1		4
1	:	16	=	0		1

$327_{10} = 147_{16}$

316	:	16	=	19	остаток	12 (c)
19	:	16	=	1		3
3	:	16	=	0		3

$327_{10} = 1x33c_{16}$

Паспорт экзамена

по дисциплине «Языки программирования», 2 семестр

1. Методика оценки

Экзамен проводится письменной форме, по билетам. Список билетов приведен ниже. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Языки программирования»

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую.
(Целые и дробные числа). Обратный и дополнительный код.
2. Аргументы, используемые по умолчанию в C++
3. 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.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)

(дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет *0-19 баллов*.
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-

следственные связи явлений, при решении задачи допускает непринципиальные ошибки, например, вычислительные, оценка составляет *20-27 баллов*.

- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает ошибок при решении задачи, оценка составляет *28-35 баллов*.
- Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет *36-40 баллов*.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к экзамену по дисциплине «Языки программирования»

Билет 1

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую.

(Целые и дробные числа). Обратный и дополнительный код.

2. Аргументы, используемые по умолчанию в C++

3. 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. Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки

3. 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. Встроенные функции

3. 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. Операция ::, перегруженные функции

3. 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. Определение переменных, константные значения в C++ (отличие от C)

3. 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. Объектно-ориентированное программирование (Инкапсуляция, наследование, полиморфизм). Понятие класса. Отличие от структур в Си.

3. 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: 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. Имена-этикетки в enum, struct и union , анонимные объединения в C++
 3. Написать на языке "C" или "C++" 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 C. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.
 2. Операторы распределения памяти в C++
 3. Написать на языке "C" или "C++" 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. Конструкторы и деструкторы
 3. Написать на языке "C" или "C++" 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. Функции друзья

3. 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. Перегрузка операций

3. 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 шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки

3. 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. Производные классы. Множественное наследование

3. 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. Полиморфизм. Виртуальные функции члены. Абстрактные классы

3. 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 u #endif)

2. Потоки (ввод-вывод в C++). Предопределенные объекты-потоки. Форматирование потоков.

(функции-элементы, флаги, манипуляторы)

3. 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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. Обработка исключений
3. 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
3. 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 $N_k = 0 \Rightarrow \text{НОД} = N_{k-1}$

($N=23345 \quad M=9135 \Rightarrow 1015 \quad N=238 \quad M=347 \Rightarrow 34$)

Билет 20

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

2. Перегрузка операций

3. 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 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. Аргументы, используемые по умолчанию в С++

3. 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. Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки
3. Написать на языке "C" или "C++" 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
// 678.99000
// -34.000000
// 0.000400
// Записано чисел: 4
//

Билет 23

1. Операторы языка. Составной оператор. Управляющие конструкции языка.
2. Встроенные функции
3. 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 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. Операция ::, перегруженные функции

3. 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[] = "111111", 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. Определение переменных, константные значения в C++ (отличие от C)

3. 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. Объектно-ориентированное программирование (Инкапсуляция, наследование, полиморфизм). Понятие

класса. Отличие от структур в Си

3. 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. Имена-этикетки в enum, struct и union, анонимные объединения в C++

3. 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. Операторы распределения памяти в C++

3. 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. Конструкторы и деструкторы

3. 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.

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

Билет 30

1. Структуры, объединения, перечисления

2. Функции дружья

3. 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. Перегрузка операций

3. 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.

x

$$\text{tg}(x) = \frac{x}{1 - \frac{x^2}{3} + \frac{x^4}{5} - \frac{x^6}{7} + \dots}$$

x²

1 -

x²

3 -

x²

5 -

7 -

Билет 32

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

под массивы. Инициализация указателей

2. Написать шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки

3. 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.

$$\begin{aligned} &1 \\ \text{sqrt}(2) &= 1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \dots \end{aligned}$$

Билет 33

1. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.
2. Написать шаблон функции – нахождение максимального значения. Шаблоны классов. Преимущества и недостатки
3. 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.

$$\begin{aligned} &2 \\ \epsilon &= 2 + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \dots \end{aligned}$$

Билет 34

1. Функции в C. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.
2. Производные классы. Множественное наследование
3. 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.

Билет 35

1. Операторы передачи управления (break, continue, goto, return)
 2. Полиморфизм. Виртуальные функции члены. Абстрактные классы
 3. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main()
- Прототип: void f(int n);
Печать римских цифр
1. I- I 10-X
 2. II- II 50-L
 3. III- III 100-C
 4. IV- IV 500-D
 5. V- V 1000-M
 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. Потоки (ввод-вывод в C++). Предопределенные объекты-потоки. Форматирование потоков.
(функции-элементы, флаги, манипуляторы)
 3. Написать на языке "C" или "C++" 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. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
 3. 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. Обработка исключений
 3. 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-операций равнялся 100. Имеется 8 решений. Распечатать их со знаками.
 $1*2+3*4*5-6+7+8-9=10$
-

Билет 39

1. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные).
Приоритеты и порядок выполнения.
 2. Шаблоны классов и функций. Написать шаблон функции – нахождение минимального значения
 3. 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. Перегрузка операций
 3. 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
 $327 : 2 = 163 \text{ остаток } 1$
 $163 : 2 = 81 \text{ } 1$
 $81 : 2 = 40 \text{ } 1$
 $40 : 2 = 20 \text{ } 0$
 $20 : 2 = 10 \text{ } 0$
 $10 : 2 = 5 \text{ } 0$
 $5 : 2 = 2 \text{ } 1$
 $2 : 2 = 1 \text{ } 0$
 $1 : 2 = 0 \text{ } 1$
 $327_{10} = 101000111_2$
-

Билет 41

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

2. Аргументы, используемые по умолчанию в C++

3. 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

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

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

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

$$\mathbf{0.125_{10} = 0.001_2}$$

Билет 42

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

2 Ссылки, параметры-ссылки, функция, возвращающая значение типа ссылки

3 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 цифр от 0 до 7. В main() массив распечатать

Перевести из десятичной системы в восьмеричную целое число

$$327 : 8 = 40 \text{ остаток } 7$$

$$40 : 8 = 5 \text{ } 0$$

$$5 : 8 = 0 \text{ } 5$$

$$\mathbf{327_{10} = 0507_8}$$

Билет 43

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

2. Встроенные функции

3. 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 шестнадцатеричную целое число

$$327 : 16 = 20 \text{ остаток } 7$$

$$20 : 16 = 1 \text{ } 4$$

$$1 : 16 = 0 \text{ } 1$$

$$\mathbf{327_{10} = 147_{16}}$$

$$316 : 16 = 19 \text{ остаток } 12 \text{ (c)}$$

$$19 : 16 = 1 \text{ } 3$$

$$3 : 16 = 0 \text{ } 3$$

$$\mathbf{327_{10} = 1x33c_{16}}$$

Паспорт расчетно-графического задания (работы)

по дисциплине «Языки программирования», 2 семестр

1. Методика оценки

Первая часть:

Разработать класс, представляющий заданный тип данных (ТД). Класс должен включать в себя (если это необходимо и подходит по смыслу к ТД):

- Конструкторы (по умолчанию, с параметрами, копии);
- Деструктор;
- Функцию вывода на экран;
- Перегруженные операторы:
 - Оператор присваивания;
 - Бинарные арифметические (+, -, *, /) с операндами ТД и ТД, ТД и базовый тип, базовый тип и ТД;
 - Арифметика с накоплением (+=, -=);
 - Унарные (++, --) в префиксной и постфиксной форме;
 - Логические (<, >, ==, !=);
 - Операторы взятия элемента ([] или ()) по номеру или ключу;
 - Операторы преобразования типа к любому базовому.

Написать головную программу, обеспечивающую проверку всех реализованных свойств и методов класса.

Требования к заданию (невыполнение каждого из требований – минус один балл):

- Проект должен состоять из следующих модулей:
 - «точка входа» в приложение с функцией `int main()`;
 - класс, описывающий заданный ТД;
 - класс, реализующий проверку функций для ТД на основе действий пользователя в консоли.
- Исходный текст программы должен быть аккуратно отформатирован.
- Проект не должен содержать глобальных переменных и общедоступных членов данных класса.

Вторая часть:

Разработать шаблон заданной структуры хранения данных (контейнера). Шаблон должен содержать функции добавления, удаления, взятия элемента по индексу, поиска элементов.

Модернизировать головную программу, обеспечивающую проверку всех реализованных свойств и методов шаблона, используя его для хранения объектов класса, разработанного в первой части РГЗ.

Критерии оценки

- Работа считается **не выполненной**, если выполнены не все части РГЗ(Р), отсутствует анализ исходных данных, не разработан алгоритм и не выполнена

программная реализация, оценка составляет 0-12 баллов.

- Работа считается выполненной на **пороговом** уровне, если части РГЗ(Р) выполнены формально: исходных данных, разработан алгоритм и выполнена программная реализация, но результат работы программы не соответствует исходной постановке задачи или программа работает некорректно, оценка составляет 13-18 баллов.
- Работа считается выполненной на **базовом** уровне, если анализ объекта выполнен в полном объеме, разработан алгоритм и выполнена программная реализация, но программа работает с небольшими недочетами, оценка составляет 19-23 баллов.
- Работа считается выполненной на **продвинутом** уровне, если анализ объекта выполнен в полном объеме, разработан алгоритм и выполнена программная реализация, программа соответствует поставленной задаче и работает без ошибок, оценка составляет 22-25 баллов.

2. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

3. Примерный перечень тем РГЗ(Р)

Варианты ТД к первой части:

1. Строка текста.
2. Дата.
3. Время.
4. Матрица произвольной размерности.
5. Суперточные вычисления (дробь).
6. Супердлинные вычисления (число в строковой записи).
7. Супердлинные вычисления (число – массив байт).
8. Полином.
9. Разреженная матрица. Ненулевые элементы хранятся в виде динамического массива записей (i, j, value).
10. Разреженная матрица. Ненулевые элементы хранятся в виде списка.

Варианты контейнеров ко второй части:

1. Динамический массив.
2. Двусвязный список.
3. Двусвязный циклический список.
4. Очередь на основе списка.
5. Стек на основе списка.
6. «Двусторонний» массив.
7. Бинарное дерево.