

**ГОУ ВПО Российско-Армянский (Славянский)
университет**

Утверждено
Директор Института математики и информатики
Арамян Р.Г.



«21» апреля 2025, протокол №9.1

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Наименование дисциплины: Архитектура ЭВМ и язык ассемблера

Авторы: канд.тех.наук, доцент Овакимян Анна Седраковна
 канд.тех.наук Акопян Дживан Андраникович

Направление подготовки: 01.03.02 Прикладная математика и информатика
Наименование образовательной программы 01.03.02 Прикладная
математика и информатика

1. АННОТАЦИЯ

1.1. Краткое описание содержания данной дисциплины;

В рамках курса студенты будут изучать основы работы ЭВМ и язык ассемблера для архитектуры (8086).

После удачного прохождения курса студенты должны:

- Знать основы организации ЭВМ, язык ассемблера 8086, принцип работы прерываний, основы организации памяти, причины появления уязвимостей в программах, физические основы работы ЭВМ.
- Уметь писать программы на языке ассемблера, анализировать причины ошибок при работе с памятью, анализировать способы улучшения производительности программ.
- Владеть навыками разработки приложений на уровне языка ассемблера.

1.2. Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет)

216 часов- 6 ЗЕТ

1.3. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)

Курс является базовым для ряда последующих курсов: Операционные системы, программирование в среде Linux.

1.4. Результаты освоения программы дисциплины:

Код компетенции	Наименование компетенции	Код индикатора достижения компетенций	Наименование индикатора достижений компетенций
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	1	Знать методы и технологии разработки и применения системного и прикладного программного обеспечения
		2	Разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения

		3	Владеть способностью разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения
ПК-8	способностью приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности	1	Знать основные методы координации деятельности органов управления, организации надзора, контроля и информационного обеспечения профессиональной деятельности
		2	Уметь применять нормативноправовые основы действующего законодательства в области решения профессиональных задач
		3	Владеть навыками принятия управленческих решений в области профессиональной деятельности
ПК-9	способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы	1	Знать план выполняемой работы в ходе практической деятельности для получения профессиональных умений и опыта профессиональной деятельности

		2	Уметь составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы
		3	Регулярно оценивать и анализировать результаты выполненной работы с целью выявления достижений, а также выявления возможных областей для улучшения и развития профессиональных навыков.
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	1	Знает, как осуществлять поиск, критический анализ и синтез информации для решения поставленных профессиональных задач
		2	Умеет применять системный подход на основе поиска, критического анализа и синтеза информации для решения задач в профессиональной области
		3	Владеет навыками поиска, синтеза и критического анализа информации в своей

			профессиональной области; владеет системным подходом для решения поставленных задач
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	1	Знает основные задачи и области применения математических методов, основные принципы математического моделирования, методы построения и анализа математических моделей
		2	Умеет выбирать математические методы, адаптировать и использовать их для разработки и реализации алгоритмов решения прикладных задач
		3	Осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач

2. УЧЕБНАЯ ПРОГРАММА

2.1. Цели и задачи дисциплины

Цель данного курса научить студентов основам организации ЭВМ, объяснить, как взаимодействуют между собой разные части ЭВМ, как программа, написанная на языке программирования, выполняется на ЭВМ.

2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах)

Виды учебной работы	Всего, в акад. часах	Распределение по семестрам					
		— сем	II сем	— сем	— сем.	— сем	— сем.
1	2	3	4	5	6	7	8
1.Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:	216						
1.1. Аудиторные занятия, в т. ч.:	64		64				
1.1.1.Лекции	32		32				
1.1.2.Практические занятия, в т. ч.	32		32				
1.2. Самостоятельная работа, в т. ч.:	80		80				
1.3. Другие методы и формы занятий	36		36				
Итоговый контроль (Экзамен, Зачет, диф. зачет - указать)			Экзамен				

2.3. Содержание дисциплины

2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

Разделы и темы дисциплины	Всего (ак. часов)	Лекции (ак. часов)	Практ. занятия (ак. часов)	Семина- ры (ак. часов)	Лабор. (ак. часов)	Другие виды занятий (ак. часов)
1	2=3+4+5 +6+7	3	4	5	6	7
Модуль 1.						
Введение	2	2	0			
Раздел 1. Архитектура набора команд, производительность и безопасность программ	38					
<i>Тема1. Регистры процессора IA32, размещение переменных в памяти. Компиляция и компоновка программ.</i>	4	2	2			

<i>Тема 2.</i> Основные арифметические команды, регистр флагов, переполнение, команды изменения естественного порядка выполнения программы	10	2	8			
<i>Тема 3.</i> Реализация управляющих операторов языка Си: условная передача данных, организация циклов.	4	2	2			
<i>Тема 4.</i> Оператор switch, организация таблиц перехода	4	2	2			
<i>Тема 5.</i> Представление программы в памяти, организация вызовов функций, фрейм функции в стеке.	4	2	2			
<i>Тема 6.</i> Указатели и агрегатные типы данных. Размещение массивов в памяти	4	2	2			
<i>Тема 7.</i> Реализация многомерных и многоуровневых массивов	4	2	2			
<i>Тема 8.</i> Размещение структур в памяти.	4	2	2			
Раздел 2. Система программирования (языка Си)						
<i>Тема 9.</i> Двоичный интерфейс приложения, соглашение о вызовах. Рекурсивные функции. Функция main.	4	2	2			
<i>Тема 10.</i> Процессор x86-64, различные соглашения о вызовах.	4	2	2			
<i>Тема 11.</i> Организация динамической памяти. Функция malloc.	4	2	2			
<i>Тема 12.</i> Переполнение буфера, эксплуатация ошибок, механизмы защиты.	4	2	2			
<i>Тема 13.</i> Числа с плавающей точкой. Представление вещественных чисел.	4	2	2			
<i>Тема 14.</i> Процессор x87. Архитектура и основные команды.	4	2	2			
<i>Тема 15.</i> Компиляция. Схема работы ассемблера.	4	2	2			

<i>Тема 16.</i> Статическая компоновка программ. Формат объектных файлов ELF. Создание статической библиотеки.	4	2	2			
<i>Тема 17.</i> Динамические (разделяемые) библиотеки.	4	2	2			
<i>Тема 18.</i> Прерывания, процессы, виртуальная память	4	2	2			
Модуль 2						
Раздел 3. Архитектура ЭВМ						
<i>Тема 19.</i> Аппаратура ЭВМ. Физические основы, логические вентили, полусумматор, полный сумматор, АЛУ.	4	2	2			
<i>Тема 20.</i> Статическая память. Защелки. Синхронная SR-защелка. Синхронная D-защелка. Шеститранзисторная ячейка статической двоичной памяти (бит) SRAM	4	2	2			
<i>Тема 21.</i> Динамическая память. Интегральные схемы. Закон Мура. Оперативная память. Энергонезависимая память.	2	2	0			
<i>Тема 22.</i> Иерархия памяти. Кэш память	4	2	2			
<i>Тема 23.</i> Организация конвейерных вычислений. Архитектуры RISC и CISC	2	2	0			
ИТОГО	64	32	32			

2.3.2. Краткое содержание разделов дисциплины в виде тематического плана

Модуль 1

Введение

Раздел 1 Архитектура набора команд, производительность и безопасность программ

Тема 1. Регистры процессора IA32, размещение переменных в памяти. Компиляция и компоновка программ.

Регистры процессора 8086, 8, 16, 32, 64 битные регистры. Размещение программы в памяти, сегменты кода, стека, кучи. Директивы, представление отрицательных чисел.
В.И. Юров [1-3]

Тема 2. Основные арифметические команды, регистр флагов, переполнение, команды изменения естественного порядка выполнения программы

Арифметические инструкции, первая программа, знаковое и беззнаковое переполнение, регистр флагов, изменение потока управления программы с учетом регистра флагов. Отображение арифметических команд языка С на язык ассемблер.

В.И. Юров [6-8]. Bryant, O'Hallaron Computer Systems. A programmer perspective [3]

Тема 3. Реализация управляющих операторов языка Си: условная передача данных, организация циклов.

Инструкции перехода. Организация условных и безусловных переходов. Организация циклов. Отображения конструкций управления потока языка С в язык ассемблер.

Bryant, O'Hallaron Computer Systems. A programmer perspective [3]

Тема 4. Оператор switch, организация таблиц перехода

Отображение оператора switch языка С в язык ассемблер. Таблицы переходов.

Bryant, O'Hallaron Computer Systems. A programmer perspective [3]

Тема 5. Представление программы в памяти, организация вызовов функций, фрейм стека функции

Организация вызовов функций. Соглашения о вызовах. Передача аргументов. Адрес возврата. Фрейм функции в стеке. Возврат значения. Пролог и эпилог функций.

Bryant, O'Hallaron Computer Systems [3].

Тема 6. Указатели и агрегатные типы данных. Размещение массивов в памяти

Работа с памятью и с указателями на уровне ассемблера. Размещение массивов в памяти. Доступ к элементам массива на уровне ассемблера.

Bryant, O'Hallaron Computer Systems [3].

Тема 7. Реализация многомерных и многоуровневых массивов

Размещение многомерных и многоуровневых массивов в памяти. Доступ к элементам матрицы. Отображения с языка С. Производительность при обращении к элементам матрицы.

Bryant, O'Hallaron Computer Systems [3]

Тема 8. Размещение структур в памяти.

Организация доступа к полям структуры, выравнивание структур. Отображения с языка С.

Bryant, O'Hallaron Computer Systems [3]

Раздел 2. Система программирования (языка Си)

Тема 9. Двоичный интерфейс приложения, соглашение о вызовах. Рекурсивные функции. Функция main.

Двоичный интерфейс приложения. Организация рекурсии на уровне ассемблера. Размещение программ в памяти. Вызов функции main.

Bryant, O'Hallaron Computer Systems [7]

Тема 10. Процессор x86-64, различные соглашения о вызовах.

64 битные регистры. Различные соглашения о вызовах. Передача аргументов через регистры. Организация вызов функций в разных ОС.

Bryant, O'Hallaron Computer Systems [7]

Тема 11. Организация динамической памяти. Функция malloc.

Реализация функции malloc. Структура кучи. Алгоритмы выделения памяти в куче.

Bryant, O'Hallaron Computer Systems [7]

Тема 12. Переполнение буфера, эксплуатация ошибок, механизмы защиты

Объяснения причин уязвимости программ при наличии ошибки переполнения буфера. Эксплуатация ошибок на тестовых программах. Механизмы защиты на уровне компиляторов и ОС (канарейка стека, ASLR, non-executable memory). Технологии безопасной разработки кода.

Bryant, O'Hallaron Computer Systems [9]

Тема 13. Числа с плавающей точкой. Представления для вещественных чисел.

Представление вещественных чисел по стандарту IEEE-754. Обновление необходимости существования стандарта. Основные арифметические операции над двойничным представлением вещественных чисел.

В.И. Юров. Ассемблер [17]

Тема 14. Процессор x87. Архитектура и основные команды.

Регистры процессора x87. Основные команды. Пересылка данных между основным процессором и сопроцессором x87.

В.И. Юров. Ассемблер [17]

Тема 15. Компиляция. Схема работы ассемблера.

Этапы компиляции программ. Промежуточное представление программ. Оптимизирующие компиляторы. Генерация кода. Распределение регистров.

Bryant, O'Hallaron Computer Systems [9]

Тема 16. Статическая компоновка программ.

Статическая компоновка программ. Разбор формата объектных файлов ELF. Создание статической библиотеки. Положительные и отрицательные стороны статических библиотек.

Bryant, O'Hallaron Computer Systems [7]

Тема 17. Динамические (разделяемые) библиотеки.

Создание динамических библиотек. Размещение динамических библиотек в памяти. Компоновка программ с динамическими библиотеками. Код, не зависящий от адреса (PIC).

Bryant, O'Hallaron Computer Systems [7]

Тема 18. Прерывания, процессы, виртуальная память.

Введение в ОС. Организация виртуальной памяти. Процессы, многозадачность. Разные виды прерываний. Системные вызовы.

Bryant, O'Hallaron Computer Systems [8, 9]

Тема 19. Аппаратура ЭВМ. Физические основы, логические вентили, полусумматор, полный сумматор, АЛУ

Принцип работы транзисторов. Логические вентили (not-and, not-or, and, or, not). Построение логических сдвигов. Схема XOR. Построение полусумматоров, полный сумматор. Мультиплексоры и декодеры. Построение АЛУ.

Bryant, O'Hallaron Computer Systems [4]

A. Tanenbaum. Structured Computer Organization [3]

Тема 20. Статическая память. Защелки.

Физические основы построения бита памяти. Синхронная SR-защелка. Синхронная D-защелка. Шеститранзисторная ячейка статической двоичной памяти (бит) SRAM. Принцип работы ЦПУ.

Bryant, O'Hallaron Computer Systems [4]

A. Tanenbaum. Structured Computer Organization [3]

Тема 21. Динамическая память.

Интегральные схемы. Закон Мура. Оперативная память. Энергонезависимая память.

Bryant, O'Hallaron Computer Systems [4]

A. Tanenbaum. Structured Computer Organization [3]

Тема 22. Иерархия памяти. Кэш память

Пирамидная иерархия памяти. Организация кэш памяти. Прямой кэш. Ассоциативный кэш. Влияние кэш памяти на производительность программ. Разработка дружелюбного к кэшу кода (cache friendly code).

Bryant, O'Hallaron Computer Systems [4]

A. Tanenbaum. Structured Computer Organization [3]

Тема 23. Организация конвейерных вычислений. Архитектуры RISC и CISC

Последовательное вычисление. Использование конвейеров. Простаивание конвейера. Предсказание ветвления. Ограничения при конвейерных вычислениях.

Bryant, O'Hallaron Computer Systems [4]

B. Tanenbaum. Structured Computer Organization [3]

2.3.3. Краткое содержание семинарских/практических занятий/лабораторного практикума

Составление программ на языке ассемблера.

Прямые задачи: Написать эквивалентную данному фрагменту на языке Си программу на языке ассемблера (для разных конструкций языка Си).

Обратные задачи: восстановить фрагмент Си кода из данной ассемблер-программы.

Организация вызовов функций с использованием разных соглашений о вызовах.

Создание и использование статических и динамических библиотек.

Работа с данными в формате с плавающей точкой.

Работа в среде SASM.

2.3.4. Материально-техническое обеспечение дисциплины

Компьютеры с доступом к интернету
Среда программирования SASM
Платформа e-judge

2.4. Модульная структура дисциплины с распределением весов по формам контролей

Формы контролей	Вес формы (форм) текущего контроля в результирующей оценке текущего контроля (по модулям)		Вес формы промежуточного контроля в итоговой оценке промежуточного контроля		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей (семестровой оценке)		Вес результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля	
	M1	M2	M1	M2	M1	M2				
Вид учебной работы/контроля	M1	M2	M1	M2	M1	M2				
Контрольная работа (<i>при наличии</i>)	0.5	0.5								
Весы результирующих оценок текущих контролей в итоговых оценках промежуточных контролей						0.5				
Весы оценок промежуточных контролей в итоговых оценках промежуточных контролей						0.5				
Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5			
Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5			

Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0.4
Вес итогового контроля (Экзамен/зачет) в результирующей оценке итогового контроля								0.6
	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$

3. Теоретический блок

3.1. Материалы по теоретической части курса

3.1.1. Учебники

1. Randal E. Brayan, David R. O'Hallaron. Computer Systems: Programmers perspective. Prentice Hall.
2. Рендел Брайант, Давид О'Халларон. Компьютерные системы: Архитектура и программирование. Взгляд программиста. Пер. сангл., СПб.: БХВ, 2005.
3. В.И. Юров Assembler. Учебник для вузов. 2-е изд., СПб.: Питер, 2003.
4. С.В.Зубков. Assembler для DOS, Windows, и Unix. Язык низкого уровня. М., ДМК, 2000г.
5. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд.,- СПб.: Питер, 2013.

3.1.2. Учебные пособия

1. Е.А. Кузьменкова, В.С. Махнычев, В.А. Падарян. Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 1. Издание 2-е, дополненное. М., Изд-во МГУ, МАКС Пресс, 2014. – 80 с.
2. Е.А. Кузьменкова, В.А. Падарян, М.А. Соловьев Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 2. М., изд-во МГУ, МАКС Пресс, 2014. – 100 с.

3.1.3. Электронные материалы: Презентации лекционного материала

3.1.4. Глоссарий/терминологический словарь;

3.1.5. др. варианты материалов, необходимых для освоения учебной программы дисциплины.

Варианты решенных задач

4. Фонды оценочных средств

4.1. Планы практических и семинарских занятий

N	Тема	Дата
1.	Двоичные числа и коды: прямой, обратный, дополнительный код. Примеры преобразования. Сложение кодов. Интервалы значений, которые можно хранить в памяти (8 бит, 16 бит ...).	
2.	Архитектура IA 32. Основные регистры. Директивы определения данных и константы. Секции data, bss, text. Знакомство с средой SASM. Организация ассемблерной программы.	
3.	Ввод/вывод. Простые арифметические команды (add, sub, mul, imul, div, idiv). Регистр флагов Привидение типов. Примеры отображения Си-программ на язык Ассемблера.	
5.	Битовые и байтовые операции. Логические команды и команды сдвига	
4.	Условное выполнение (if-else, switch). Циклы.	
5.	Указатели и массивы. Представление массивов.	
6.	Прямые и обратные задачи на массивы и указатели.	
7.	Структуры. Правила выравнивание. Переход с Ассемблера на С.	
8.	Функции. Передача аргументов. Рекурсии. Фрэйм функции. соглашения: cdecl, fastcall, stdcall. Нарушение calling convention, Передача структур в функцию. Вызов функций по указателю.	
9.	Задачи с вызовами стандартных библиотечных функций (abs, scanf, printf, malloc). Чтения и запись в файлы) .	
10.	Числа с плавающей точкой. Представления для вещественных чисел. Процессор x87. Архитектура и основные команды.	

4.2. Материалы по практической части курса

4.2.1. Учебно-методические пособия

1. Е.А. Кузьменкова, В.С. Махнычев, В.А. Падарян. Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 1. Издание 2-е, дополненное. М., Изд-во МГУ, МАКС Пресс, 2014. – 80 с.
2. Е.А. Кузьменкова, В.А. Падарян, М.А. Соловьев Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 2. М., изд-во МГУ , МАКС Пресс, 2014. – 100 с.

4.3. Вопросы и задания для самостоятельной работы студентов

Задачи для самостоятельного выполнения в среде e-judge (15 задач к первому модулю, 19 задач ко второму модулю)

4.4. Образцы вариантов контрольных работ, тестов и/или других форм текущих и промежуточных контролей

Вариант 1. Выписать значение регистра AL в виде десятичного числа (знакового и беззнакового), а также флаги CF, OF, ZF и SF после выполнения следующих инструкций.

задача	решение	Объяснение
MOV AL, 199 ADD AL, -61	al = 138 eflags [cf, sf]	$ \begin{array}{r} 199 = 11000111 \\ -61 = 11000011 \text{ (195)} \\ \hline 110001010 \\ 138 = 10001010 \end{array} $
MOV AL, -35 SUB AL, 216	al = 5 eflags [pf]	$ \begin{array}{r} -35 = 11011101 \text{ (221)} \\ 216 = 11011000 \\ \hline 5 = 000000101 \end{array} $
MOV AL, -13 ADD AL, 179	al = 166 eflags [cf, pf, sf]	$ \begin{array}{r} -13 = 11110011 \text{ (243)} \\ 179 = 10110011 \\ \hline 166 = 10100110 \end{array} $
MOV AL, 64 ADD AL, 82	al = 146 eflags [of, sf]	$ \begin{array}{r} 64 = 01000000 \\ 82 = 01010010 \\ \hline 146 = 10010010 \end{array} $

Вариант 2. Выписать значение регистра AL в виде десятичного числа, а также значение флага CF после выполнения следующих битовых инструкций.

Команды	Ответ
mov al, 10 shl al, 6	al = -128 CF = 0
mov al, 10 shl al, 5	al = 64 CF = 1
mov al, 10 shr al, 1	al = 5 CF = 0
mov al, 10	al = 2

shr al, 2	CF = 1
mov al, -10 shr al, 1	al = 123 CF = 0
mov al, -10 sal al, 1 mov al, 127 sal al, 1	al = -20 CF = 1 al = -2 CF = 0, OF = 1
mov al, -10 sar al, 2	al = -3 CF = 1
mov al, 10 rol al, 4	al = -96
mov al, 10 rol al, 5	al = 65 CF = 1
mov al, 10 ror al, 1	al = 5
mov al, 10 rcl al, 5	al = 64 CF = 1
mov al, 10 rcr al, 2	al = 2 CF = 1

Вариант 3. Прямые задачи.

Перевести фрагмент C- программы на язык Ассемблера:

a) static short int a = 10;

static int b = 20, c;

c = a + b;

b) unsigned char z = 0xff;

unsigned short a = 0xff00;

unsigned int b;

b = a * z;

c) static int a, b = 1, c = -2, d = 3; ...

a = b + c * d;

```

d) static int a;

    short b = 1;

    int c = -2, d = 3, e = 4;

    a = b - c * d * e;
e) void cond(int a, int *p) {
    if (p && a > 0) {
        *p += a;
    }
}

```

Вариант 4. Обратные задачи. Восстановить Си-код из ассемблер-кода:

```

a) section .bss
    a resd 1
section .data
    b dd 1
    c dw -2
    z dw 5
section .text
    mov bx, word[z]
    mov ax, word[c]
    imul bx
    mov ecx, dword[b]
    movsx eax, ax
    sub ecx, eax
    mov dword [a], ecx

b) Восстановить Си-функцию из ассемблер-кода:
    mov eax, dword [ebp + 8] ; параметр x
    mov edx, dword [ebp + 12] ; параметр y
    cmp eax, -3
    jge l2
    cmp eax, edx
    gle l3
    imul eax, edx
    jmp l4
l3:
    add eax, edx
    jmp l4
l2:
    cmp eax, 2
    jg l5
    xor eax, edx

```

```

        jmp l4
l5:
        sub eax, edx
l4:
Замечание: Си-функция задана с пропущенными выражениями в операторах
int test(int x, int y) {
    int val = _____;
        if (_____) {
            if (_____) {
                val = _____;
            } else {
                val = _____;
            }
        } else if (_____) {
            val = _____;
        }
    return val;
}

```

Вариант 5. Задачи на программирование на языке Ассемблера.

1. Вычислить значение выражения: $y = a - x + b/c$
2. Найти ребро прямоугольника, если известны площадь и другое ребро.
3. Вычислить площадь треугольника, если даны (x, y) координаты трех вершин.
4. Даны две статические переменные a и b размером 2 байта. Требуется регистр EAX заполнить следующим образом: верхние 2 байта должны содержать значение переменной a, а нижние 2 байта – значение переменной b, циклически сдвинутое на 5 бит вправо.
5. Вычислить значение выражения $y = 32 * x - x / 8 + x \% 16$ с использованием команд сдвига.
6. Найти самую правую единицу в двоичном представлении числа.
7. Подсчитать число символов введенной строки, включая последний символ – перевод строки.
8. Подсчитать сумму n чисел.
9. Вычислить сумму арифметической прогрессии. С клавиатуры задаются: первый член прогрессии, шаг, количество членом прогрессии, которые требуется суммировать.
10. В заданном массиве int a[10] найти максимальный и минимальный элементы.
11. Дан массив из 100 элементов типа int. Найти сумму элементов массива, превосходящих значение последнего элемента.
12. Даны два массива из 100 элементов типа short. Проверить массивы на равенство и напечатать 1 в случае положительного ответа и 0 в противном случае.

4.5. Перечень экзаменационных вопросов

1. Представление целых и вещественных чисел в двоичной форме. Прямой, обратный и дополнительный коды. Формат IEEE-754. Порядок представления байтов в памяти (big and little endian). Размеры типов в языке C. Знаковые и беззнаковые

числа. Приведение типов.

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective, вторая глава.
- Книга Зубков С. В, первая глава
- Книга В.И. Юров: глава 4, глава 17

2. Процессор x86, регистры. Формат данных. Пересылка данных (mov). Регистр eflags. Арифметические и логические операции. Сдвиги. Переполнение. Флаги OF, CF, ZF. Отображение кода из языка C в язык ассемблера и наоборот.

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective: 3.1 - 3.6
- Книга В.И. Юров: глава 7, 8, 9

3. Флаги OF, CF, ZF, SF. Команды передачи управления. Инструкция jmp и условный переход. Представление операторов условного перехода языка Си на языке ассемблера. Оператор switch.

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective: 3.1 - 3.6
- Intel reference manual
- Книга В.И. Юров: глава 10

4. Организация циклов. Инструкция loop. Представление операторов цикла языка Си в языке ассемблера (do-while, while, for). Инструкция условной пересылки (Conditional Move Instructions)

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective: 3.1 - 3.6
- Книга В.И. Юров: глава 10

5. Представление программы в памяти. Стек и локальные переменные. Расположение глобальных и статических переменных. Организация вызова функций. Передача аргументов, адрес возврата. Фрейм стека. Различные соглашения о вызовах.

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective: 3.7
- Книга Зубков С. В, 5.2, 5.3
- Книга В.И. Юров: глава 15

6. Массивы и указатели. Представление одномерных и двумерных массивов. Структуры данных в ассемблере.

Литература:

- Книга R. Brayan. Computer Systems Programmers perspective: 3.1 - 3.6
- Книга В.И. Юров: глава 13

7. Процессор x87. Организация процессора и его регистры. Основные команды процессора x87.

Литература:

- Книга В.И. Юров: Глава 17

8. RISC-V

Литература:

- <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf>
- <https://github.com/riscv/riscv-bitmanip/releases/download/1.0.0/bitmanip-1.0.0.pdf>

4.6. Образцы экзаменационных билетов

Образец 1

1. 1. Процессор x86, регистры. Регистр eflags. Арифметические операции. Флаги OF, CF, ZF. Отображение из C в ассемблер и наоборот.
2. 2. Организация памяти. Кеш-память. Реализация кеш-памяти. Локальность кода.
3. 3. Задача.

Образец 2

4. Прямой, обратный и дополнительный коды. Порядок представления байтов (big and little endian). Знаковые и беззнаковые числа в языке C и Ассемблера. Приведение типов.
5. Цифровой логический уровень организации процессоров. Вентили, декодеры и мультиплексоры. Полусумматор и полный сумматор. АЛУ.
6. Задача.

4.7. Образцы экзаменационных практических заданий

1. Вычислить площадь треугольника, если даны (x, y) координаты трех вершин.
2. Вычислить значение выражения $y = 32 * x - x / 8 + x \% 16$ с использованием команд сдвига.
3. Найти самую правую единицу в двоичном представлении числа.
4. Подсчитать число символов введенной строки, включая последний символ – перевод строки.

4.8. Банк тестовых заданий для самоконтроля

Кодирование чисел: Представить следующие числа в прямом, обратном и дополнительном коде: 8, 9, -9, -12

	прямой	обратный	дополнительный
8	00001000	00001000	00001000
9	00001001	00001001	00001001
-9	10001001	11110110	11110111
-12	10001100	11110011	11110100

Система команд: Пересылка данных (mov, movsx, movzx). Определить значение регистра после операций:

- mov al, 10

- `mov al, 127`
- `mov bh, 300 // bh = 44, bx = 11264`
- `mov ax, 0xffff // ax = 0xffff`
`mov al, 1 // ax = 0xff01, al = 1, ah = 0xff`
- `mov bx, 45`
`movzx ax, bl`
- `mov bx, -45 // bx = 0xffd3`
`movzx ax, bl // ax = 0xd3`
- `mov bl, -1 // bl = 0xff, bx = 0xff`
`movsx ax, bl // ax = 0xffff`

5. Методический блок

5.1. Методика преподавания

Преподавание учебной дисциплины «Архитектура ЭВМ и язык Ассемблера» строится на сочетании лекций, практических занятий и различных форм самостоятельной работы студентов.

На лекциях рассматриваются основные вопросы по темам, составляющим содержание дисциплины, с акцентом на наиболее сложные проблемы.

На практических занятиях делается акцент на проверку теоретических знаний и их применению при решении практических задач и программированию на языке NASM в среде программирования SASM.

Рабочей программой дисциплины предусмотрена самостоятельная работа студентов, включающая работу с литературными и электронными ресурсами, выполнение домашних заданий в среде e-judge, подготовку рефератов.

Промежуточный контроль проводится с использованием платформы e-judge.