

Интро

План: один пишет другой проверяет добавляет новое/дополняет

Шаблон:

Написал: Никита А.


Проверил: Никита К.

Машинное слово это бла-бла-бла

Источники информации:

- Лекции на гитлаб

<https://git.iu7.bmstu.ru/alexodnodvorcev/swampofdespair/-/tree/master/Sem>

- Стандарт C99  ISO_IEC_9899_TC3.pdf
- IEEE 754 - ЧПТ

Правила:

- Оставлять ссылку к источнику и примерное нахождение в нём, т.е.

пример:

Источник: преза 1.3.2 6 слайд, что эквивалентно sem1.3.2 6 слайд.

или: Источник: C99 2.5.7

Вопросы

1. Из каких частей состоит абстрактная машина, с которой мы работаем? На основе какой машины она построена?

Написал: Никита А.

Проверил: Виталик Г.

Источник: Sem 1.01.1 Об устройстве ЭВМ

Части абстрактной машины (Принстонская архитектура)

Центральный процессор, ЦП

Некоторое Арифметико-логическое устройство, которое управляет загрузкой/выгрузкой данных памяти и оборудованием.

Оперативное запоминающее устройство, ОЗУ

Предоставляет место для хранения данных и кода, в виде одномерного массива ячеек, к каждой из которых процессор может обратиться по адресу.

Система ввода-вывода, IO

Позволяет процессору общаться с внешним оборудованием, включая оборудование для взаимодействия с пользователем. Жёсткий диск тоже внешнее устройство, с которым процессор общается через IO.

2. Дайте определение машинному слову. Попробуйте предположить, как будет действовать машина с машинным словом, равным 4 (5, 6, 7, 8) байтам, в ответ на запрос передачи 3 байт информации между компонентами. Какова причина возникновения машинного слова?

Написал: Никита А.

Проверили: Виталик Г. Инкина Ю.

Источник: Sem 1.01.1 Об устройстве ЭВМ [слайд 8]

Машинное слово. Сообщение фиксированного размера для коммуникации между процессором и другими частями ПК. Точное значение машины неизвестно, мы будем выбирать между 4 и 8 байтами. (Размер слова в битах различается на различных архитектурах, например на intel 8086 имел длину слова 16 бит. Для совместимости с программами для этого процессора в следующую версию процессора добавили возможность исполнять слова с длиной $\frac{1}{2}$ слова длиной 32 бита и так далее до современных чипов, в которых до сих пор можно исполнять очень старый код)

Причина: "Процессор не может общаться с другими частями ПК с помощью сообщений в бит, накладные расходы такого общения были бы больше полезной информации."

Машинное слово не может быть передано как 3 байта, так как оно не соответствует размеру машинного слова, поэтому, изначально в запросе будет пустота (`\x00`).

3. Дайте определения трансляции и транслятору.

Написал: Никита А.

Проверил: Виталя Г.

Источник: Sem 1.01.1 Об устройстве ЭВМ

Трансляция. Перевод кода с языка высокого уровня на язык машины.

Транслятор. Программа-переводчик с языка высокого уровня на язык машины.

Препроцессором. Программа, осуществляющая препроцессинг, комплекс действий над текстом на языке высокого уровня перед основной трансляцией.

Компоновщик. Программа, преобразующая набор транслированных модулей в единую программу.

Дополнительно важно знать порядок компиляции программы на языке C:

Порядок компиляции в Си

1. Препроцессинг

Command ``gcc -E source.c -o preprocessed.i``, где `source.c` - исходный код, `preprocessed.i` - файл после препроцессинга.

В него входит:

- убираются комментарии,
- добавляются заголовочные файлы (которые тоже проходят препроцессинг)

- Происходит замена макросов, таких как ``#define``, ``#include``.

2. Трансляция

Command ``gcc -S preprocessed.i -o object_file.s``
(.asm)

Трансляция языка программирования в Ассемблер.
(машинный код, понятный человеку)

На этом этапе мы еще можем прочесть результат и понять.

3. Ассемблирование (Компиляция, **трансляция**)

Команда: `gcc -c preprocessed.i(или .s , если из ассемблера)`
`-o object_file.o`

Может заменять прошлый пункт и сразу из ``.i`` файла создавать файл с машинным кодом понятным для компьютера.
(not human-readable) и отдельно от пункта 2, и после.`

4. Компоновка

Команда: `gcc object_file.o -o app.exe`

Бывает 2х видов, т.е.:

Первый с использованием библиотек уже находящихся в системе, т.е. Добавление ссылок на них в исполняемый файл.
[Static and Dynamic Libraries | Set 1 - GeeksforGeeks](#)

Второй с добавлением каждой библиотеки в исполняемый файл, т.е. добавление бинарников библиотек в сам исполняемый файл, из-за чего размер исполняемого файла больше. [Working with Shared Libraries | Set 1 - GeeksforGeeks](#)

5. Загрузка (вызов загрузчика для загрузки в память, возможен вызов динамических библиотек на этом этапе)

Команда: ``./app.exe``

4. Дайте определение времени трансляции и времени выполнения.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.2 Линейная программа 2-4 слайд

Время трансляции.

Время когда программа-переводчик выполняет перевод с языка высокого уровня на язык машины.

Отсылка к Си: объектный файл создается после трансляции,

Объектный файл. Файл в котором расположен машинный код написанной нами программы, который зачастую не невозможно запустить. Т.к. Используя директиву `include` мы добавляем библиотеки, которые не прописаны в этом коде.

Время выполнения.

Время когда процессор выполняет код на машинном языке.

5. Дайте определение интерпретатору и компилятору.

Написал: Никита А.

Проверил: Никита К.

Источник: <https://nuancesprog.ru/p/12524/>

Интерпретатор.

Программа которая построчно выполняет трансляцию во время исполнения.

Компилятор.

Программа которая, заранее перед выполнением, производит препроцессинг, потом трансляцию в результате которой генерируется объектный файл. Далее создаёт новый, исполняемый, файл в который линкуются (добавляются ссылки) или прописываются используемые библиотеки.

6. Дайте определение слову языка программирования.

Написал: Никита А.

Проверил: Никита К.

Источник: ОТСУТСТВУЕТ

Некоторая лексема влияющая на поведение программы.

Подробнее:

Лексема единица информации, которая используется для описания программы.

В контексте программирования, слова могут быть ключевыми словами, идентификаторами, литералами, операторами или комментариями

7. Дайте определение зарезервированному слову языка программирования.

Написал: Никита А.

Проверил: Никита К.

Источник: ОТСУТСТВУЕТ

Лексема влияющая на последовательность выполнения действий программы.

Зарезервированное слово не может быть использовано в качестве идентификатора.

Или же: слово, которое заранее определено в языке программирования и используется для выполнения определенных функций или задач.

8. Можно ли в программе на языке Си дать переменной имя «if» («IF», «cos», «If», «return», «whiLE»)?

Написал: Виталик Г.

Проверил: Никита А.

6.4.1 Keywords

Syntax

keyword: one of

auto	enum	restrict	unsigned
break	extern	return	void
case	float	short	volatile
char	for	signed	while
const	goto	sizeof	_Bool
continue	if	static	_Complex
default	inline	struct	_Imaginary
do	int	switch	
double	long	typedef	
else	register	union	

Semantics

The above tokens (case sensitive) are reserved (in translation phases 7 and 8) for use keywords, and shall not be used otherwise. The keyword **_Imaginary** is reserved specifying imaginary types.⁵⁹⁾

В программе есть ключевые слова времени препроцессинга и обычные, сначала идет обработка препроцессором, на этом этапе может быть не выявлено что вы использовали ключевые слова как переменные, и даже может быть использовано ключевое слово как переменная, с дальнейшей ее заменой препроцессором. Но на этапе компиляции это выдаст ошибку.

Итог:

- if -
- IF +
- cos + (можно перезаписать)

- If +
- return -
- whILE +

Источник C99 стр. 145

```

-----
preprocessing-file:
    groupopt

group:
    group-part
    group group-part

group-part:
    if-section
    control-line
    text-line
    # non-directive

if-section:
    if-group elif-groupsopt else-groupopt endif-line

if-group:
    # if      constant-expression new-line groupopt
    # ifdef   identifier new-line groupopt
    # ifndef  identifier new-line groupopt

elif-groups:
    elif-group
    elif-groups elif-group

elif-group:
    # elif    constant-expression new-line groupopt

else-group:
    # else    new-line groupopt

endif-line:
    # endif  new-line

control-line:
    # include pp-tokens new-line
    # define  identifier replacement-list new-line
    # define  identifier lparen identifier-listopt )
                    replacement-list new-line
    # define  identifier lparen ... ) replacement-list new-line
    # define  identifier lparen identifier-list , ... )
                    replacement-list new-line
    # undef   identifier new-line
    # line    pp-tokens new-line
    # error   pp-tokensopt new-line
    # pragma  pp-tokensopt new-line
    #          new-line

text-line:
    pp-tokensopt new-line

non-directive:
    pp-tokens new-line

lparen:
    a ( character not immediately preceded by white-space

replacement-list:
    pp-tokensopt

pp-tokens:
    preprocessing-token
    pp-tokens preprocessing-token

new-line:
    the new-line character

```

9. Дайте определение точке входа и точке выхода.

Написал: Никита А.

Проверил: Виталик Г.

Источник: Sem 1.01.2 Линейная программа

Точка входа. Точка входа – строка, с которой начинается исполнение программы, будь она интерпретируемой, а не компилируемой.

Точка выхода. Строка, после которой исполнение программы прекращается.

10. Может ли у приложения отсутствовать точка входа?
Может ли она быть не единственна?

Написал: Никита А.

Проверил:

Она **ЕДИНСТВЕННА**.

Точка входа **ДОЛЖНА БЫТЬ**, исключение библиотека (**НЕУБЕРЕН**).

11. Может ли у приложения отсутствовать точка выхода? Может ли она быть не единственна?

Написал: Никита А.

Проверил: Виталик Г.

Источник: Sem 1.01.2 Линейная программа

Может отсутствовать, если это бесконечный цикл.

В Си точек выхода у приложения может быть несколько.

12. Как можно найти точку входа в проекте на языке Си?

Написал: Никита А.

Проверил: Виталик Г.

По реализации прототипа функции: `"int main()"` (верно, Костёр на 1 семе говорил про это)

13. Препроцессор языка Си работает до этапа трансляции или после?

Написал: Никита А.

Проверил: Виталик Г.

До этапа трансляции.(смотреть 3 вопрос)

14. Какие команды препроцессора Вы знаете?

Написал: Никита А., Юля И.

Проверил:

Источник: c99 6.10 Preprocessing directives c145-162

(Постараюсь сделать вменяемый перевод нескольких позже)

Равно директивам препроцессора.

`#define`

Обозначает константу, которая будет прописана в код при препроцессинге, либо некую функцию с тем же смыслом.

`#include`

Обозначает другой файл который будет включен в текущий.

"a preprocessing translation unit."

"A `#include` preprocessing

directive causes the named header or source file to be processed from phase 1 through phase 4, recursively. All preprocessing directives are then deleted."

15. Что определяет язык программирования? Может ли транслятор не поддерживать стандарт?

Написал: Виталик Г.

Проверил:

Источник: 1 лекция 1 сема проги по питону

Поддержка стандарта:

16. Дайте определение стандартной библиотеке.

Написал: Виталик Г.

Проверил:

Источник: Вики библиотека и здравый смысл.

Стандартная библиотека C представляет собой стандартизированную коллекцию библиотек, используемых для реализации общих операций.

17. Какие модули стандартной библиотеки Вы уже знаете?

math.h

stdio.h

stdlib.h

18. Можно ли давать свои макроопределения математических констант, которые уже есть в стандартной библиотеке?

Можно, это макросы, а именно `#define`

19. Можно ли не давать макроопределение константы, которая используется в программе более двух раз?

Написал: Никита А.

Проверил:

Источник: подход разработчика

Можно не давать, но плохой тон, т.к. В значении при написании можно сделать ошибку, либо константа может измениться в какой-то момент

20. Дайте определение типу данных.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайд 2

Тип данных. Пара из множества значений и операций между ними.

21. Дайте определение простому типу данных.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайд 2

Простой тип данных. Если никакая сущность данного типа не включает в себя сущность простого типа.

22. Обязана ли машина обеспечивать аппаратную поддержку какого-либо типа данных?

Написал: Виталик Г.

Проверил:

Нет, машина может вообще не иметь специального сопроцессора для вычисления математики. А значит и операций для этого типа. Тогда реализация этого типа может лежать на плечах программиста.

23. Перечислите целочисленные типы в Си.

Написал: Никита А.

Проверил: Виталик Г.

Источник: Sem 1.01.3 Простые типы данных слайды 3-12

Short

Int

Long

Long long

Char

Unsigned - семейство беззнаковых типов данных

Size_t - беззнаковый тип, означающий адрес в памяти

[u]int[size in bites]_t: int8_t, int32_t,

Uint32_t - целых фиксированного размера

24. Каков размер каждого из целочисленных типов в Си? Дайте оценку максимальному числу в каждом из них.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайды 3-12

Источник: c99 стр. 21-22

"The values given below shall be replaced by constant expressions suitable for use in #if preprocessing directives. Moreover, except for CHAR_BIT and MB_LEN_MAX, the following shall be replaced by expressions that have the same type as would an expression that is an object of the corresponding type converted according to the integer promotions. **Their implementation-defined values shall be equal or greater in magnitude (absolute value) to those shown, with the same sign.**"

Short: more than 2 bytes, до бесконечности

Int: between short and long

Long: more than 4 bytes

Long long: more than 8 bytes

Char: 1 byte

Unsigned: не указано (зависит от предка)

Size_t: не превышает размер адреса в памяти машины

Unsigned ...: информации нет, предполагаю, что то же что и у их предков

[u]int[size in bites]_t: информации нет, предполагаю, что то же что и у их предков

25. Как ведут себя переменные каждого из целых типов при переполнении?

Написал: Никита А.

Проверил: Виталий Г.

Источник: Sem 1.01.3 Простые типы данных слайды 3-12

Переполнение беззнаковых происходит по кольцу, т.е. После максимального числа будет 0, после 0 - 1, будет максимальное число.

Переполнение знаковых является ситуацией неопределённого поведения (undefined behavior), т.е. неизвестно что произойдет.

26. Опишите концепцию опциональной особенности, обязательной при имплементации, на примере фиксированных целых в Си.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайд 13

Данная концепция означает, что типы могут НЕ ПОДДЕРЖИВАТЬСЯ компилятором, но если поддерживаются, то ОБЯЗАТЕЛЬНО должны следовать стандарту.

27. Перечислите типы чисел с плавающей точкой (ЧПТ) в Си. Связаны ли они с типами с теми же названиями в других языках? Опишите их связь со стандартом IEEE 754.

Написал: Никита А.

Проверил: Никита К.

Источник: Sem 1.01.3 Простые типы данных слайды 13

Float, double, long double.

Различные ЯП:

Типы чисел с плавающей точкой могут иметь разные размеры и точности в разных языках программирования, но они обычно связаны с другими языками, имеющими те же названия для этих типов.

IEEE 754:

ИСТОЧНИК YANDEXGPT: все следуют и все отличаются от стандарта.

Стандарт IEEE 754 определяет формат представления чисел с плавающей точкой, включая мантиссу и показатель степени, а также специальные значения, такие как бесконечность и NaN (Not a Number - "не число")

28. Можно ли сравнивать два ЧПТ через «двойное равно» (1)? Как правильно сравнивать два ЧПТ (2)? Как правильно сравнивать два ЧПТ, исходя из предположения, что все используемые ЧПТ близки к единице по модулю (3)?

Написал: Никита А.

Проверил: Виталик Г.

Источник: лаба по Си

- (1) Можно ли сравнивать через `==`: нет
- (2) Как правильно сравнивать: через Эпсилон до 10^8 степени, т.е. разницу чисел умножить на эпсилон и сравниваем с нулём.
- (3) Не уверен что ПОЛНОСТЬЮ понял этот вопрос, но видимо речь о сравнении мантииссы, так как оба числа близки к 1. А значит мы можем пропустить сравнение экспоненты

29. Дайте определение значащему разряду.

Написал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайды 16

Приведение типа. Изменение типа данных явно или не явно, вызвано невозможностью выполнить какую либо операцию без изменения типа.

Как работает приведение. В большинстве случаев, у транслятора в приоритете минимизация потерь, т.е. Целые приводятся к ЧПТ

Значащий разряд. Количество ненулевых в конце у ЧПТ.

30. Найдите значение выражения: $1e-12 + 1e+12$.

Написал: Никита А.

Проверил:

Источник: компилятор

Ответ: $1e+12$

Сохранение максимальной точности, начиная с целой части.

31. Найдите значение выражения: $1e-12 - 1e+12$.

Написал: Никита А.

Проверил:

Источник: компилятор

Ответ: $-1e+12$

Сохранение максимальной точности, начиная с целой части.

32. Найдите значение выражения: $1e-16 + 1e-17$.

Написал: Никита А.

Проверил:

Источник: компилятор

Ответ: $1.100000e-16$

33. Найдите значение выражения: $1e+64 + 1e+64$.

Сделал: Никита А.

Проверил: Виталик Г.

Источник: компилятор

Ответ:

%le:

2.000000e+64

34. Могут ли машина и человек предполагать разную точность у полученного в ходе вычислений ЧПТ?

Double с точность float, т.е. в некоторых случаях в переменной double может находиться double с точностью float числа.

35. Найдите значение выражения: $3.1234567891234 + 2.1f$.

Сделал: Никита А., Юля И.

Проверил: Виталик Г.

Источник: компилятор, **с99 стр. 278**

Ответ: 5.223457

Почему 6 знаков?

(f,F) If the precision is missing, it is taken as 6; if the precision is zero and the # flag is not specified, no decimal-point character appears

(g,G) 6 if the precision is omitted

36. Опишите используемый Вами алгоритм выбора типов для решения задачи.

1. Первым требуется осуществить выбор между целыми типами и ЧПТ. Если в задаче указано количество штук, или алгоритм предполагается целочисленным, то перед Вами определённо целое. Если у сущности можно представить наличие размерности, то скорее пригодится ЧПТ.
2. По умолчанию в качестве типа ЧПТ выбираем число двойной точности, если экономия четырёх байт не играет роли, иначе – число одинарной точности.
3. Если уверены, что выигрыш в ширине диапазона в два раза спереди необходим, причём невозможно выбрать знаковое большего объёма, то беззнаковое, иначе – знаковое.
4. Если указано в требованиях строгое соответствие размеру, то фиксированное целое. Если переменная используется в качестве логического флажка, кода возврата, или необходима максимальная близость к машине, то скорее подойдёт `int`. Если нужен самый большой диапазон, то `long long`.
5. Если буферная переменная для массива произвольной длины, то скорее `char`.
6. Никогда `short`. :-)
7. Во всех остальных случаях по вкусу – например, `long`.

37. Как работают явное и неявное приведения типов в Си?

Сделал: Никита А.

Проверил:

Источник: Sem 1.01.3 Простые типы данных слайд 16

Явное приведение. Приведение вызванное программистом с помощью круглых скобок перед переменной и типа, к которому приводим.

Неявное приведение. Приведение вызванное невозможностью компилятора провести операцию, например сложение инта с даблом неопределенно, поэтому необходимо приведение инта к даблу. Определяется таблицей, компилятор стремится минимизировать потерю точности, так-что `int` приводится к `double`.

38. Правда ли, что в выражении $(4 \leq 3.9)$ транслятор сначала приведёт тип значения левого операнда к `double`?

Написал: Никита А.

Проверил:

Источник:

Да, приведёт, к `double`

39. Можно ли поменять тип у переменной в языке Си?

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3 page 16

Си - язык со статической типизацией. Тип переменной менять нельзя, но можно поменять тип выражения.

40. Дайте определение переменной. Попробуйте отталкиваться от знания составных частей абстрактной машины и свойств переменной, которых у неё не может не быть.

Написал: Никита А.

Проверил: Виталик Г.

Источник: ОТСУТСТВУЕТ

Поименованная ссылка на область данных в памяти машины определенного типа данных.

41. Опишите концепцию статической типизации.

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3 page 16

Си - язык со статической типизацией. Тип переменной менять нельзя, но можно поменять тип выражения.

42. Опишите тип данных `int`.

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3 page 5

Приближен к конкретной машине, размер не указан.

`sizeof(short) <= sizeof(int) <= sizeof(long)`

Спецификатор ввода-вывода: ```%d```

Замкнут относительно / и %.

Диапазон значений: от $-2^{(N-1)}$ до $2^{(N-1)}$, где N количество бит в размере

Поведение переполнения неопределенно.

Тип дабл

1e40

Мантисса: 16 десятичных разрядов, остальное мусор

43. Дайте определение замкнутому относительно операции A типу данных.

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3 page 14

Тип данных, при выполнении над которым операции A останется тем же типом данных.

Пример: ЧПТ замкнуты относительно любых операций над ними же, (благодаря специальным значениям Inf , $-Inf$, NaN). Все целые типы замкнуты относительно $\%$ и $/$, беззнаковые еще замкнуты относительно $*$, $+$, $-$

44. Замкнут ли тип `int` относительно целочисленного деления? Относительно сложения?

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3

Да, тип `int` замкнут относительно `/` и не замкнут относительно остальных, если это обычный `int`, если это `unsigned int`, то он замкнут относительно `+`.

45. Замкнут ли тип `double` относительно умножения?
Деления?

Написал: Виталик Г.

Проверил: Никита А.

Источник: sem 1.01.3

ЧПТ замкнуты относительно любых операций над ними же,
(благодаря специальным значениям `Inf`, `-Inf`, `NaN`).

Дополнительная секция вопросов

1. $1e64 * 1e64$?

Написал: Никита А.

Проверил:

Источник:

$1e128$, меньше $1e308$

2. $1e300 * 1e300$?

Написал: Никита А.

Проверил:

Источник:

Inf, т.к. Больше 308

Константа DBL_MAX_10_EXP c99 28p., 308p. (point 4)

3. $1e-300 * 1e-300$

Написал: Никита А.

Проверил:

Источник:

0.00000

4. Что такое Inf и Nan

Написал: Никита А.

Проверил:

Источник: c99 23p.

Inf.

По факту `inf` это бесконечность в математике, при том, что мы считаем, что если число больше `1e308` это бесконечность.

The usual mathematical formulas for complex multiply, divide, and absolute value are problematic because of their treatment of infinities and because of undue overflow and underflow.

A character sequence `INF` or `INFINITY` is interpreted as an infinity, if representable in the return type, else like a floating constant that is too large for the range of the return type.

Nan.

По факту `nan` это любая неопределённость, мы же не знаем, `inf * 0` или `inf - inf` это что-то неопределённое, как вывод это неопределённость

A NaN is an encoding signifying Not-a-Number. A quiet NaN propagates through almost every arithmetic operation without raising a floating-point exception; a signaling NaN generally raises a floating-point exception when occurring as an arithmetic operand.

A character sequence `NAN` or `NAN(n-char-sequenceopt)`, is interpreted as a quiet NaN, if supported in the return type, else like a subject sequence part that does not have the expected form; the meaning of the n-char sequences is implementation-defined.²⁵⁹) A pointer to the final string is stored in the object pointed to by `endptr`, provided that `endptr` is not a null pointer.

5. `sqrt(-1)?`

Написал: Никита А.

Проверил:

Источник: c99 ~24p.

NaN, чтоб не провоцировать исключение невозможности исполнения

6. `Inf * 0.0?`

Написал: Никита А.

Проверил:

Источник: компилятор

nan

7. `inf/inf?`

Написал: Никита А.

Проверил:

Источник: компилятор

nan

8. `1 / 0?`

Написал: Никита А.

Проверил:

Источник: c99 94p.

The result of the `/` operator is the quotient from the division of the first operand by the second; the result of the `%` operator is the remainder. In both operations, if the value of the second operand is zero, the **behavior is undefined.**

9. $1.0 / 0.0$?

Написал: Никита А.

Проверил:

Источник:

`inf`

10. $1e+5 / 1e-307$

Написал: Никита А.

Проверил:

Источник:

`Inf`, т.к. $1e+312 > 1e+308$, экспонента не вмещается, получается `inf`

11. $1e-300 / 1e+307$?

Написал: Никита А.

Проверил:

Источник:

`0.000000e+00`, потому, что меньше $1e-308$, дальше некуда, он становится чистым `0`.

46. Составьте «таблицы неявного приведения» для четырёх арифметических операций над числами типов int, long int, long long int, float, double, int32_t, int64_t.

Написал: Виталик Г.

Проверил:

Источник: sem 1.01.3

	int	long	lli	float	double	int32_t	int64_t
int	int						
long	long	long					
lli	lli	lli	lli				
float	float	float	float	float			
double	double	double	double	double	double		
int32_t	int32_t	long	lli	float	double	int32_t	
int64_t	int64_t	int64_t	lli	float	double	int64_t	int64_t

	int	long int	long long int	float	double	int32_t	int64_t
int	int	long int	long long int	float	double	int32_t/int	int64_t
long int	long int	long int	long long int	float	double	long int	int64_t
long long int	long long int	long long int	long long int	float	double	long long int	long long int/int64_t
float	float	float	float	float	double	float	float
double	double	double	double	double	double	double	double
int32_t	int32_t/int	long int	long long int	float	double	int32_t	int64_t

int64_t	int64_t	int64_t	long long int/int64_t	float	double	int64_t	int64_t
---------	----------------	----------------	---------------------------------------	--------------	---------------	----------------	----------------

47. Может ли целое число служить логическим выражением?

Написал: Виталик Г.

Проверил: Никита А.

Да. Доказательство:

...

```
int a = 1; // logical flag
if (a) // здесь число 1 это выражение
{
    printf("something");
}
...
```


48. Какое целое число всегда ложно? Только ли оно?

Написал: Виталик Г.

Проверил:

Других ложных чисел, кроме нуля, нет

49. Определен ли порядок арифметических вычислений?

50. Укажите порядок выполнения операций: $a + b + c + a + c + d + b$

Написал: Никита А.

Проверил:

: $((a + b) + (c + a)) + (c + d) + b$

Ваня: поэлементно слева право, слова Ломовского

Лево ассоциативная операция, но порядок не определен

51. Укажите порядок выполнения операций: $f(a) + f(b) + f(f(a))$

Написал: Никита А.

Проверил:

Источник:

Левосторонний порядок операций, т.е.

1. $f(a)$
2. $f(b)$
3. $f(a) + f(b)$
4. $f(a)$
5. $f(f(a))$
6. $(f(a) + f(b)) + f(f(a))$

52. Можем ли мы так писать и почему: `rc = scanf("%d", &a) + scanf("%d", &b);`

Сделал: Никита А., Виталик Г.

Поведение не определено.

53. Можем ли мы так писать и почему: `rc = scanf("%d%d", &a, a);`

Написал: Никита А.

Отточник: мысли

Нельзя, т.к. в `scanf` вторым и далее аргументом принимаются только адреса, тем более `"%d"` означает тип `*int`.

54. Каковы особенности вычислений с ЧПТ, в частности, будет ли оптимизировано вычисление:

$x = y - y; z = y / y;$

Написал: Никита А.

Проверил:

Источник: преза типы данных

Оптимизации нет, не факт, что результат будет нулем.

55. Найдите значение суммы 1039 слагаемых:

$$1e-40 + 1e-40 + \dots + 1e-40 + 1.0$$

Сделал: Никита А.

Проверил:

Источник: семинар Александра Сергеевича Кастрицкого

Ответ: 1.000000e+00

Точнее сказать ровно 1.0, т.к. существует вещь под названием отбрасывание мантииссы.

Наводящие на понимание вопросы:

1. $1e-40 + 1e-40$? ($2e-40$)
2. $1e-24 + 1e-40$? ($1e-24$), т.к. больше 16 разрядов разницы экспоненты и мантиисса отбрасывается
3. $1e-01 + 1e-24$? ($1e-01$), т.к. больше 16 разрядов разницы экспоненты и мантиисса отбрасывается

Экспонента числа с плавающей точкой. Часть с $1e-N$, где N это число экспоненты, идет в первых разрядах числа.

Мантиисса числа с плавающей точкой. Фактически само дробное число, т.е. $Ae-01$, A – мантиисса.

56. Определен ли порядок вычисления «аргументов» (левого и правого операндов) логических И/ИЛИ?

Написал: Никита А.

Источник: мысли

Сначала вычисляются левый аргумент, далее с слева на право.

Н0! Это не всегда так, поэтому АС может сказать поведение не определено. Не пытайтесь с ним спорить.

57. Сколько существует булевых функций двух (трёх, n) аргументов?

58. Опишите концепцию ленивого вычисления логических выражений в языке Си.

Сделал: Никита А.

Проверил: Никита К.

Источник: опыт,

<https://tproger.ru/translations/10-useful-things-in-c>

(4 пункт)

В случае когда исход логического выражения уже известен оно не выполняется до конца. Примеры:

```
int t1() return 1;  
int t2() return 1;  
int f1() return 0;  
int f2() return 0;
```

```
t1() && t2() || f1() && f2()
```

Т.к. в первой части выражения (t1() && t2()) мы получаем истину, и далее идёт операнд истины

59. Присутствует ли ленивая логическая схема в других языках? Можно ли на неё полагаться при написании программы на Си?

Написал: Никита А.

Проверил:

Источник: мысли

В python существует, в принципе в python, да

60. Можем ли мы так писать и почему:

```
if(scanf("%d", &x) == 1 && x > 0)
```

Да, можем, из-за последовательности выполнения команд, сначала выполнится `scanf`, только потом, если `scanf` считывает число, тогда произойдёт проверка на `x`.

61. Можем ли мы так писать и почему:

```
if(scanf("%d", &x) && x > 0)
```

Написал: Никита А.

Проверил:

Источник: мысли

С одной стороны да, можем из-за последовательности выполнения, мы не можем пойти дальше, пока не вычислим результат выполнения `scanf`.

С другой нет, данный код выглядит нелогично, т.к. мы можем получить в качестве результата `-1`, если введем `0EF`, а в переменной `x` у нас будет мусор и проверка будет неверная.

62. Можем ли мы так писать и почему:

```
rc = scanf("%d", &a);
```

```
rc = rc || scanf("%d", &b);
```

Написал: Никита А.

Проверил:

Ответ нельзя, почему не знаю.

```
```c
 int a, b;
 printf("\n%d %d \n", a, b);
 int rc = scanf("%d", &a);
 printf("\n%d %d \n", rc, a);
 rc = rc || scanf("%d", &b);
 printf("\n%d %d\n", rc, b);
```
```

Тест с введенным значением, получилось ввести только первое

```
```result
448465065 32767 // default values
4 // entered value
1 4 // result code and value
1 32767 // old result code and other value
```
```

Тест с OEF (ctr+D), получилось ввести только первое

```
```result
-144487511 32765 // default values
-1 -144487511 // result code and value
1 32765 // Why '-1' became '1' ???
```
```

63. Найдите значение выражения: $27 \ \&\& \ 13 \ \&\& \ 1 \ \&\& \ 12$.

Написал: Виталик Г.

Проверил: Никита А.

Операция И всегда возвращает либо 1 либо 0.

Это логическое и, значит если все числа не равны 0, то и результат равен 1 (True)

64. Дайте определение операторным скобкам.

Написал: Никита А.

Проверил:

Источник: АС

Операторный скобки. Определяют в языке программирования комплекс команд.

65. Дайте определение оператору.

Написал: Никита А.

Проверил:

Источник: phind

Оператор. Меняет поток выполнения программы

Выражение vs инструкции, ???

Наименьшая автономная часть языка программирования, команда.

выражение – это последовательность операций и операндов, которая указывает как вычислить значение или обозначает объект или функцию или порождает побочный эффект или выполняет комбинацию вышеперечисленного

66. Опишите синтаксис условного оператора.

Написал: Никита А.

Проверил:

Источник: Sem 1.02.4 Организация циклов слайд

```
if (bool-condition)
{
    some-operation;
} else if (other-bool-condition)
{
    other-operation;
} else
{
    lie-operation;
}
```

`bool-condition`, `other-bool-condition` - некоторое выражение, которым может быть либо истинно, либо ложно.

`some-operation`, `other-operation` - некоторая операция выполняющаяся, если выражение истинно.

`Lie-operation` - часть выполняется, если оба в данном случае выражения сложны.

67. Какие виды циклов Вы знаете? Какие ключевые слова использует каждый из них?

Написал:

Проверил:

Источник: Sem 1.02.4 Организация циклов слайд

Счётный цикл

Мы знаем сколько будет итераций

```
"for (init-state; continue-condition; update-state)"
```

Цикл с предусловием (предпроверкой)

Мы знаем при каком условии выполняется цикл

```
"while (bool-condition)"
```

Цикл с постусловием

Мы знаем при каком условии продолжается цикл

```
"do {} while (bool-condition);"
```

68. Чем отличаются выражение и инструкция? Верна ли с точки зрения синтаксиса каждая из следующих строк:

```
for (a = 2; for (a = 2; a < 54; a++); a++)
```

```
for (for (a = 2; a < 54; a++); a < 54; a++)
```

```
for (a = 2; a < 54; for (a = 2; a < 54; a++))
```

Написал: Никита А.

Проверил:

Источник: Компилятор

Всё не верно, не обозначен тип переменной a:

```
"main.c:13:10: error: 'a' undeclared (first use in this function)"
```

69. Можно ли через один цикл переписать любой другой? Если да, то через какой?

Написал: Никита А.

Проверил:

Источник: Sem 1.02.4 Организация циклов слайд 3

Все циклы можно заменить циклом с предпроверкой.

70. Дайте определение числовой последовательности.

Написал: Никита А.

Проверил:

Источник: Sem 1.03.1 Суммы последовательностей слайд 2

Числовая последовательность. Числовой последовательностью называется пронумерованный набор элементов числового множества.

71. Дайте определение рекуррентной формуле.

Написал: Никита А.

Проверил:

Источник: Сем 1.03.1 Суммы последовательностей слайд 2

Рекуррентная формула. Элементы последовательности связаны рекуррентной формой, если начиная с k элемента, все остальные зависят от всех предыдущих.

72. Дайте определение простой рекуррентной формуле.

Написал: Никита А.

Проверил:

Источник: Sem 1.03.1 Суммы последовательностей слайд 3

Простая рекуррентная формула. Рекуррентная формула считается простой если следующий элемент зависит только от текущего элемента.

73. Опишите правила чтения квантора суммы на примере:

$$s = \sum_{i=j}^k a_i.$$

Чему равна s , если $j < k$; $j = k$; $j > k$?

Написал: Никита А.

Проверил: Лёша Ш.

Источник: мысли...

Переменная i по умолчанию равна j , суммируются элементы массива a с индексами i до момента, когда $i == j$

$j < k$: элементам с индексами от j до k

$j = k$: элементу с индексом j

$j > k$: нулю

74. Можно ли подсчитать

$$\sum_{i=0}^{j:|a_j|\leq\epsilon} a_i,$$

если последовательность $\{a_i\}_i$ не является сходящейся?

Её можно посчитать, если в последовательности существует число не больше ϵ , иначе нельзя

75. Приведите пример такой последовательности, сумма из предыдущего пункта которой, посчитанная до $\varepsilon = 10^{-6}$, отличается от суммы ряда больше, чем на 10^{-6} . Можно ли оценивать остаточный член суммы с помощью ε из задачи?

76. Какие части отношения $\frac{a_i}{a_{i-1}}$ при подсчёте суммы элементов последовательности $\{a_i\}_i$ мы пытаемся сократить и почему, а какие – нет?

77. Упростите выражение:

$$\frac{(i-1)!(i+1)!}{i!(i-2)!}$$

Написал: Никита К.

Проверил: Никита А.

Источник: мысль

The image shows a handwritten solution on blue grid paper. At the top left, the number 77 is circled. To its right, the expression $(i-1)(i)(i+1)$ is written. Below this, the original fraction $\frac{(i-1)!(i+1)!}{i!(i-2)!}$ is written and underlined. A horizontal line is drawn through the fraction. Below the line, the expression $i!(i-2)!$ is written, with a diagonal line through it. To the right of the fraction, the result $= i^2 - 1$ is written. At the bottom left, there is a small handwritten mark that looks like a stylized '1' or a checkmark.

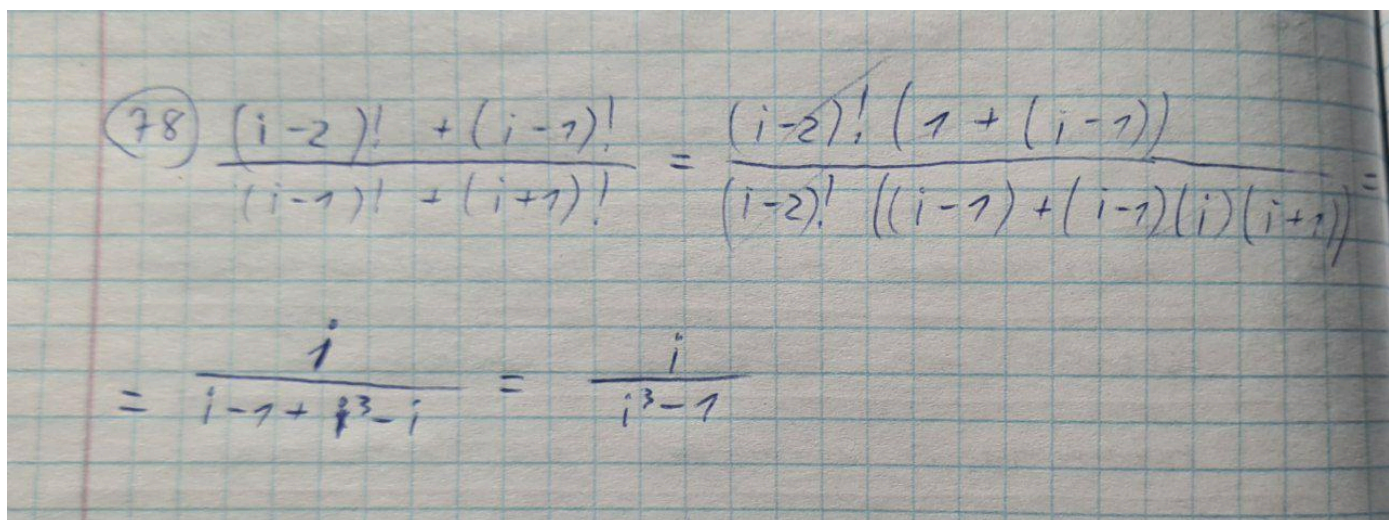
78. Упростите выражение:

$$\frac{(i-2)! + (i-1)!}{(i-1)! + (i+1)!}$$

Написал: Никита К.

Проверил: Никита А.

Источник: мысль


$$\begin{aligned} \textcircled{78} \quad \frac{(i-2)! + (i-1)!}{(i-1)! + (i+1)!} &= \frac{(i-2)! (1 + (i-1))}{(i-2)! ((i-1) + (i-1)(i)(i+1))} \\ &= \frac{i}{i-1 + i^3 - i} = \frac{i}{i^3 - 1} \end{aligned}$$

В общем, выносим самое маленькое за скобку, оно сокращается, а мы радуемся жизни

79. Опишите концепцию ввода данных по конечному признаку.

Написал: Никита А.

Проверил:

Источник: мысли

Ввод по конечному признаку означает ввод, до определённого значения, к примеру, ввод чисел по одному и вычисление промежуточного результата до OEF

В чём смысл? В том чтоб не хранить всю последовательность при некоторых операциях например нахождения максимального (см. ниже).

80. Применим ли ввод по концевому признаку к задаче нахождения максимума (минимума) среди чисел, среднего арифметического (геометрического) чисел?

Написал: Никита А.

Проверил:

Источник: мысли

Для нахождения максимума/минимума и поиска среднего арифметического (геометрического) ввод по концевому признаку применим.

В первом случае храним текущее максимальное/минимальное.
В втором случае храним сумму и количество чисел в последовательности.