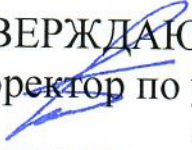


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Емельянов Сергей Геннадьевич
Должность: ректор
Дата подписания: 27.09.2024 07:47:48
Уникальный программный ключ:
9ba7d3e34c012eba476ffd2d064c075919f53be739df2374d16f3e0ce536f0fc6

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)
Кафедра вычислительной техники

УТВЕРЖДАЮ
Проректор по учебной работе

О.Г. Локтионова
« 25 » 09 2023 г.

Компьютерная арифметика

Методические указания по выполнению лабораторной работы по дисциплине «Информатика» для направлений подготовки 09.03.04, 12.03.04

УДК 681.3

Составители: Е.А. Коломиец, Т.Н. Конаныхина

Рецензент

Кандидат технических наук, доцент Л.В. Стародубцева

Компьютерная арифметика: методические указания по выполнению лабораторной работы по дисциплине «Информатика» / Юго-Зап. гос. ун-т; сост. Е.А. Коломиец, Т.Н. Конаныхина. Курск, 2023. 20 с. Библиогр.: с. 17.

Приводятся алгоритмы двоичного кодирования. Рассматривается методика выполнения арифметических операций в компьютере. Приводятся приемы использования встроенных средств табличного процессора для выполнения арифметических операций. Теоретический материал сопровождается примерами.

Методические рекомендации предназначены для студентов, обучающихся по направлениям подготовки 09.03.04, 12.03.04.

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.

Усл.печ. л. 1,16. Уч.-изд. л. 1,05.

Тираж 100 экз. Заказ 957 . Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Цель работы

Получить знания о видах двоичного кодирования в ЭВМ, методах и алгоритмах выполнения арифметических операций; научиться выполнять арифметические операции над кодами двоичных чисел; овладеть навыками использования различных инструментов, предоставляемых табличным процессором, для выполнения операций двоичной арифметики.

Краткая теоретическая информация

Способы представления двоичных чисел в компьютерной арифметике

Все современные ЭВМ имеют достаточно развитую систему команд, включающую десятки и сотни машинных операций. Однако выполнение любой операции основано на использовании простейших микроопераций типа «сложение» и «сдвиг». Это позволяет иметь единое арифметико-логическое устройство для выполнения любых операций, связанных с обработкой информации.

В принципы работы вычислительных машин заложен принцип двоичного кодирования. В таблице 1 представлены правила сложения двоичных цифр двух чисел A и B . Здесь показаны правила сложения двоичных цифр A_i и B_i одноименных разрядов $i (i \geq 0)$ с учетом возможных переносов из предыдущего разряда $P_{i-1} (P_0 = 0)$. Именно данные таблицы 1 положены в основу выполнения любой операции ЭВМ.

В обычной математике мы работаем с числами, которые могут быть и положительными, и отрицательными. При записи двоичных чисел со знаком в их формате необходимо предусмотреть два поля: поле, определяющее знак числа, и поле, характеризующее модуль числа. Под знак числа отводится специальный знаковый бит (один двоичный разряд). Остальные разряды определяют модуль числа. Знаковый разряд приписывается слева от модуля числа (старший разряд), причём знаку «+» соответствует нулевое значение знакового бита, а знаку «-» – единичное.

Кодируя не только значение числа, но и его знак, мы получаем прямой код числа.

Таблица 1

Правила сложения двоичных цифр

значения разрядов двоичных чисел A и B			значение разряда суммы	значение переноса в следующий разряд
A_i	B_i	P_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Действия над прямыми кодами двоичных чисел при выполнении операции сложения создают большие трудности, связанные с необходимостью учета значений знаковых разрядов:

- во-первых, следует отдельно обрабатывать значащие разряды чисел и разряды знака;

- во-вторых, значение разряда знака влияет на алгоритм выполнения операции.

Во всех ЭВМ все операции выполняются над числами со знаками, представленными специальными машинными кодами.

В истории развития компьютеров использовались три основных варианта представления знаковых чисел:

- прямой код или знак и величина;
- обратный код или код с дополнением до единицы;
- дополнительный код или код с дополнением до двух.

Итак, прямой код двоичного числа образуется из абсолютного значения этого числа и кода знака, например:

$$A_{10} = +12_{10} \text{ПК}(A_2) = 01100$$

$$B_{10} = -21_{10} \text{ПК}(B_2) = 110101$$

Во всех трёх кодах положительные числа выглядят одинаково.

Различия в форме записи отрицательных чисел в обратном и дополнительном кодах касаются только способа представления модуля числа, а способ кодирования и место расположения знакового бита остаются неизменными.

Обратный код получается путем инвертирования всех битов модуля (1 меняем на 0, а 0 меняем на 1).

$$A_{10} = +12_{10} PK(A_2) = 01100 \quad OK(A_2) = 01100$$

$$B_{10} = -21_{10} PK(B_2) = 110101 \quad OK(B_2) = 101010$$

Дополнительный код можно получить следующими способами. Первый:

- получить обратный код;
- к обратному коду прибавить «1» арифметическим сложением, например:

Второй:

$$A_{10} = +12_{10} PK(A_2) = 01100 \quad OK(A_2) = 01100 \quad ДК(A_2) = 01100$$

$$B_{10} = -21_{10} PK(B_2) = 110101 \quad OK(B_2) = 101010 \quad ДК(B_2) = 101011$$

Выполнить инверсию всех старших (за исключением знакового бита) битов числа кроме последней (младшей) единицы и тех нулей, которые стоят после нее, например:

$$PK(A_2) = 110101 \quad ДК(A_2) = 101011$$

Дополнительный код – наиболее распространенный способ представления отрицательных чисел.

Простейшие арифметические операции с целыми числами

При выполнении алгебраического сложения знаковый разряд и цифры модуля рассматриваются как единое целое и обрабатываются совместно. Перенос из старшего (знакового) разряда в обратном и дополнительном кодах учитывается по-разному. В случае обратного кода единица переноса из знакового разряда прибавляется к младшему разряду суммы. При использовании дополнительного кода единица переноса из знакового разряда отбрасывается.

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код согласно таблице 2.

Таблица 2

Преобразование чисел

Требуемая операция	Необходимое преобразование
$A + B$	$A + B$
$A - B$	$A + (-B)$
$-A + B$	$(-A) + B$
$-A - B$	$(-A) + (-B)$

Здесь A и B неотрицательные числа. Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. Сложение двоичных чисел осуществляется последовательно, поразрядно в соответствии с таблицей 1. При выполнении сложения чисел необходимо соблюдать следующие правила:

- слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа до знакового разряда и незначащие нули справа к дробной части числа;

- знаковые разряды участвуют в сложении так же, как и значащие.

- приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

- знак результата формируется автоматически;

- результат представляется в том коде, в котором представлены исходные слагаемые.

Числа, хранящиеся в компьютере, не могут быть сколь угодно большими и имеют некоторые предельные значения. Пусть для хранения чисел используются 8-ми разрядные регистры. Тогда при попытке записать в него число $-129_{10} = 110000001_2$ возникнет ситуация, связанная с нехваткой технического устройства для записи 9-го разряда. Такая «аварийная» ситуация называется переполнением разрядной сетки, или просто переполнением. (англ. overflow – переполнение «сверху»).

Правила сложения чисел в прямых кодах:

- складывать можно только числа с одинаковыми знаками;
- перед выполнением операции знак чисел запоминается, и складываются модули чисел (в знаковом бите записывается «0»);
- если получившаяся сумма в знаковом разряде имеет «1» («-»), то произошло переполнение, в этом случае числового значения в ответе писать не нужно;
- если получившаяся сумма в знаковом разряде имеет «0» («+»), то запомненное значение знакового разряда числа нужно восстановить, это и будет результат операции сложения.

Например, сложим числа $A_{10} = -5$ $ПК(A_2) = 10000101$ и $B_{10} = -29$ $ПК(B_2) = 10011101$ в формате 8 бит:

Числа имеют одинаковые знаки, запоминаем знак $Z_n = 1$, складываем модули

$$\begin{array}{r} + 00000101 \\ 00011101 \\ \hline 00100010 \end{array}$$

Сумма 00100010 в знаковом разряде содержит «0» («+»), восстанавливаем истинный знаковый разряд, тогда прямой код суммы это $ПК(SUM_2) = 10100010$, что соответствует $SUM_{10} = -34$.

Правила сложения чисел в обратных кодах (перед началом операции числа должны быть представлены в обратных кодах):

- дополняем каждое число битом переноса, в который заносим значение, равное значению знакового разряда числа;
- выполняем операцию сложения по правилам двоичной арифметики, не выделяя знаковый бит и бит переноса;
- если при вычислении суммы появился перенос из старшего разряда, то его нужно прибавить к числу;
- если значение бита переноса суммы не равно значению знакового разряда, то произошло переполнение, в этом случае числового значения в ответе писать не нужно;
- если значение бита переноса суммы равно значению знакового разряда, то получен обратный код суммы.

Например, сложим числа $A_{10} = -5$ $ОК(A_2) = 11111010$ и $B_{10} = -29$ $ОК(B_2) = 11100010$ в формате 8 бит. При сложении для

каждого числа введем еще один бит – бит переноса (C), начальное значение этого разряда равен значению знакового разряда числа (5):

$$\begin{array}{r} C \ S \\ + 1:11111010 \\ \underline{1:11100010} \\ 1:11011100 \end{array}$$

При сложении возник перенос из бита C . По правилам сложения в обратных кодах этот перенос нужно прибавить к числу:

$$\begin{array}{r} C \ S \\ + 1:11011100 \\ \underline{0:00000001} \\ 1:11011101 \end{array}$$

Значение бита переноса равно значению знакового бита двоичного кода суммы, следовательно, переполнения нет.

Обратный код суммы в знаковом разряде содержит «1» («←»), тогда прямой код суммы это $ПК(SUM_2) = 10100010$, что соответствует $SUM_{10} = -34$.

Рассмотрим еще один пример. сложим числа $A_{10} = -78$ $OK(A_2) = 110110001$ и $B_{10} = -59$ $OK(B_2) = 11100010$ в формате 8 бит. При сложении для каждого числа введем еще один бит – бит переноса (C), начальное значение этого разряда равен значению знакового разряда числа (S):

$$\begin{array}{r} C \ S \\ + 1:10110001 \\ \underline{1:11000100} \\ 1:01110101 \end{array}$$

При сложении возник перенос из бита C . По правилам сложения в обратных кодах этот перенос нужно прибавить к числу:

$$\begin{array}{r} C \ S \\ + 1:01110100 \\ \underline{0:00000001} \\ 1:01110101 \end{array}$$

Значение бита переноса не равно значению знакового бита двоичного кода суммы, следовательно, произошло переполнение.

Правила сложения чисел в дополнительных кодах (перед началом операции числа должны быть представлены в дополнительных кодах):

- дополняем каждое число битом переноса, в который заносим значение, равное значению знакового разряда числа;
- выполняем операцию сложения по правилам двоичной арифметики, не выделяя знаковый бит и бит переноса;
- если при вычислении суммы появился перенос из старшего разряда, то его нужно отбросить;
- если значение бита переноса суммы не равно значению знакового разряда, то произошло переполнение, в этом случае числового значения в ответе писать не нужно;
- если значение бита переноса суммы равно значению знакового разряда, то получен дополнительный код суммы.

Например, сложим числа $A_{10} = -5$ $DK(A_2) = 11111011$ и $B_{10} = -29$ $DK(B_2) = 11100011$ в формате 8 бит. При сложении для каждого числа введем еще один бит – бит переноса (C), начальное значение этого разряда равен значению знакового разряда числа (S):

$$\begin{array}{r}
 C \quad S \\
 + 1:11111011 \\
 \quad 1:11100011 \\
 \hline
 1:11011110
 \end{array}$$

При сложении возник перенос из бита C . По правилам сложения в дополнительных кодах этот перенос нужно отбросить.

Значение бита переноса равно значению знакового бита двоичного кода суммы, следовательно, переполнения нет.

Дополнительный код суммы в знаковом разряде содержит «1» («←»), тогда прямой код суммы это $PK(SUM_2) = 10100010$, что соответствует $SUM_{10} = -34$.

Рассмотрим еще один пример. Сложим числа $A_{10} = -78$ $DK(A_2) = 110110010$ и $B_{10} = -59$ $DK(B_2) = 111000101$ в формате 8 бит. При сложении для каждого числа введем еще один

бит – бит переноса (C), начальное значение этого разряда равен значению знакового разряда числа (S):

$$\begin{array}{r} C \ S \\ + 1:10110010 \\ \quad 1:11000101 \\ \hline 1:01110111 \end{array}$$

При сложении возник перенос из бита C . По правилам сложения в дополнительных кодах этот перенос нужно отбросить.

Значение бита переноса не равно значению знакового бита двоичного кода суммы, следовательно, произошло переполнение.

Задание

Вариант 1

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент сложения целых чисел A и B , представленных в прямых кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для вывода суммы в прямом коде;
- ячейки должны содержать комментарии, поясняющие их назначение;
- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;
- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Вариант 2

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент вычитания целых чисел A и B ,

представленных в прямых кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для вывода разности в прямом коде;
- ячейки должны содержать комментарии, поясняющие их назначение;
- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;
- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Вариант 3

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент сложения целых чисел A и B , представленных в обратных кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа A , преобразованного в обратный код;
- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа B , преобразованного в обратный код;
- на листе табличного процессора должна быть отведена ячейка для вывода суммы в обратном коде;
- на листе табличного процессора должна быть отведена ячейка для вывода суммы, преобразованной в прямой код;

- ячейки должны содержать комментарии, поясняющие их назначение;
- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;
- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Вариант 4

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент вычитания целых чисел A и B , представленных в обратных кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа A , преобразованного в обратный код;
- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа B , преобразованного в обратный код;
- на листе табличного процессора должна быть отведена ячейка для вывода разности в обратном коде;
- на листе табличного процессора должна быть отведена ячейка для вывода разности, преобразованной в прямой код;
- ячейки должны содержать комментарии, поясняющие их назначение;
- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;
- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Вариант 5

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент сложения целых чисел A и B , представленных в дополнительных кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа A , преобразованного в дополнительный код;
- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа B , преобразованного в дополнительный код;
- на листе табличного процессора должна быть отведена ячейка для вывода суммы в дополнительном коде;
- на листе табличного процессора должна быть отведена ячейка для вывода суммы, преобразованной в прямой код;
- ячейки должны содержать комментарии, поясняющие их назначение;
- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;
- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Вариант 6

Используя стандартные средства табличного процессора (таблица 3), разработать инструмент вычитания целых чисел A и B , представленных в дополнительных кодах.

Требования к выполнению задания.

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа A , представленного в прямом коде;
- на листе табличного процессора должна быть отведена ячейка

ка для отображения двоичного числа A , преобразованного в дополнительный код;

- на листе табличного процессора должна быть отведена ячейка для ввода двоичного числа B , представленного в прямом коде;

- на листе табличного процессора должна быть отведена ячейка для отображения двоичного числа B , преобразованного в дополнительный код;

- на листе табличного процессора должна быть отведена ячейка для вывода разности в дополнительном коде;

- на листе табличного процессора должна быть отведена ячейка для вывода разности, преобразованной в прямой код;

- ячейки должны содержать комментарии, поясняющие их назначение;

- длина числа A не должна превосходить 8 разрядов, длина числа B не должна превосходить 8 разрядов;

- если операция не может быть выполнена, должно появляться соответствующее сообщение, если в результате выполнения операции произошло переполнение, должно появиться соответствующее сообщение.

Общие требования

При оформлении работы в Excel необходимо использовать шрифт Times New Roman, размер 14, начертание обычное. Размеры ячеек выбираются по контексту, каждая ячейка должна иметь пояснения, какая информация в ней находится.

Составить отчет по результатам выполнения лабораторной работы. Отчет должен содержать:

- титульный лист (Приложение А);
- Содержание;
- Цель работы;
- Задание;
- Словесный подробный алгоритм выполнения задания с указанием функций процессора, используемых для выполнения конкретного действия;
- Скриншот интерфейса инструмента преобразования чисел;
- Скриншот листа выполнения задания с отображением формул и с отображением результатов вычислений;

- Выводы по работе.

Номера страниц проставляются внизу страницы по правому краю (шрифт – Times New Roman 12 пт). Первой страницей является титульный лист, нумерация проставляется, начиная с Цели работы отчета.

Макет Оглавления приведен в Приложении Б.

Для оформления заголовков разделов следует использовать стиль «Заголовок 1». Параметры стиля:

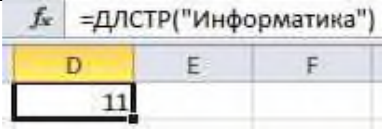
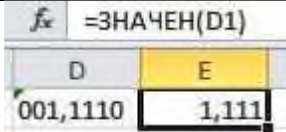
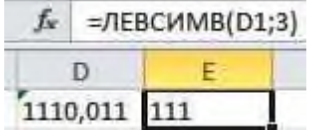
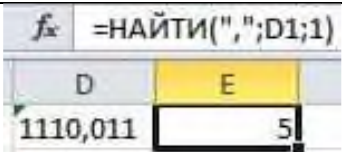
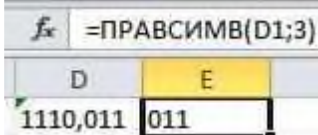
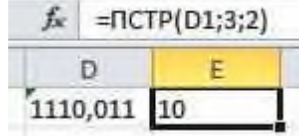
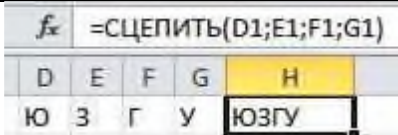
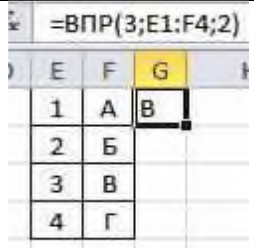
- название шрифта – Times New Roman;
- размер шрифта – 14 пт;
- междустрочный интервал – полуторный;
- интервалы перед и после абзаца – 12 пт;
- выравнивание – по ширине;
- отступы слева и справа – 0 см;
- отступ первой строки – 1,25 см.

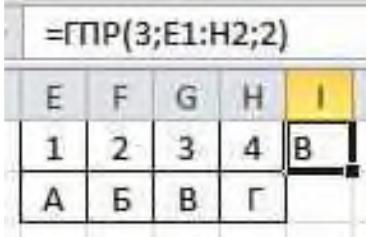
Для оформления текста отчета необходимо использовать стиль «Обычный». Параметры стиля:

- название шрифта – Times New Roman;
- размер шрифта – 14 пт;
- междустрочный интервал – полуторный;
- интервалы перед и после абзаца – 0 пт;
- выравнивание – по ширине;
- отступы слева и справа – 0 см;
- отступ первой строки – 1,25 см.

Таблица 3

Справка по функциям табличного процессора

функция	назначение	пример использования
ДЛСТР(текст)	возвращает количество знаков в текстовой строке	
ЗНАЧЕН(текст)	преобразует текстовый аргумент в число	
ЛЕВСИМВ(текст; количество знаков)	возвращает указанное количество знаков с начала строки текста	
НАЙТИ(искомый_текст; просматриваемый_текст; начальная позиция)	Возвращает позицию начала искомой строки текста в содержащей ее строке текста. Прописные и строчные буквы различаются	
ПРАВСИМВ(текст; число_знаков)	возвращает указанное число знаков с конца строки текста	
ПСТР(текст; начальная_позиция; количество_знаков)	возвращает заданное число знаков из строки текста, начиная с указанной позиции	
СЦЕПИТЬ(текст1; текст2;...)	объединяет несколько текстовых строк в одну	
ВПР(искомое_значение; таблица; номер_столбца; ...)	ищет значение в крайнем левом столбце таблицы и возвращает значение ячейки, находящейся в указанном столбце той же строки	

функция	назначение	пример использования
ГПР(искومه_значение; таблица; номер_строки; ...)	ищет значение в верхней строке таблицы и возвращает значение ячейки, находящейся в указанной строке того же столбца	

Контрольные вопросы:

1. Что такое прямой код? Как преобразовать число в прямой код?
2. Что такое обратный код? Как преобразовать число в обратный код?
3. Что такое дополнительный код? Как преобразовать число в дополнительный код?
4. Какие правила необходимо выполнять при сложении в прямом коде?
5. Какие правила необходимо выполнять при сложении в обратном коде?
6. Какие правила необходимо выполнять при сложении в дополнительном коде?
7. Что такое переполнение и когда оно возникает?

Список использованных источников

1. Колокольникова, А.И. Информатика [Текст] : учебное пособие / А.И. Колокольникова. – Москва ; Берлин : Директ-Медиа, 2020. – 289 с. : ил., табл. – Режим доступа: URL: <https://biblioclub.ru/>
2. Информатика [Текст] : лабораторный практикум / авт. – сост. О.В. Вельц. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2018. – 117 с. – Режим доступа: URL: <https://biblioclub.ru/index.php?page=book&id=494706>.

Оформление титульного листа отчета по лабораторной работе

The image shows a Microsoft Word document with a grid background. The text is as follows:

- Top center: **Минобрнауки России** (1), **ЮЗГУ** (1), **ФФиПИ** (1), кафедра наименование кафедры (2)
- Center: **ЛАБОРАТОРНАЯ РАБОТА №1** (3), **Технологии обработки текстовой информации** (4)
- Bottom right: ст. гр. XX-016 (5), Фамилия И.О. (5), Фамилия И.О. (5), Иванова Е.Н. (5)
- Bottom center: Курск 2020 г. (6)

Dimensions and margins are indicated by arrows:

- 20 mm (top and bottom)
- 30 mm (left margin)
- 15 mm (right margin)

Примечание: в кружках обозначены номера наборов параметров форматирования текста.

1: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

2: Шрифт Times New Roman, размер 16 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

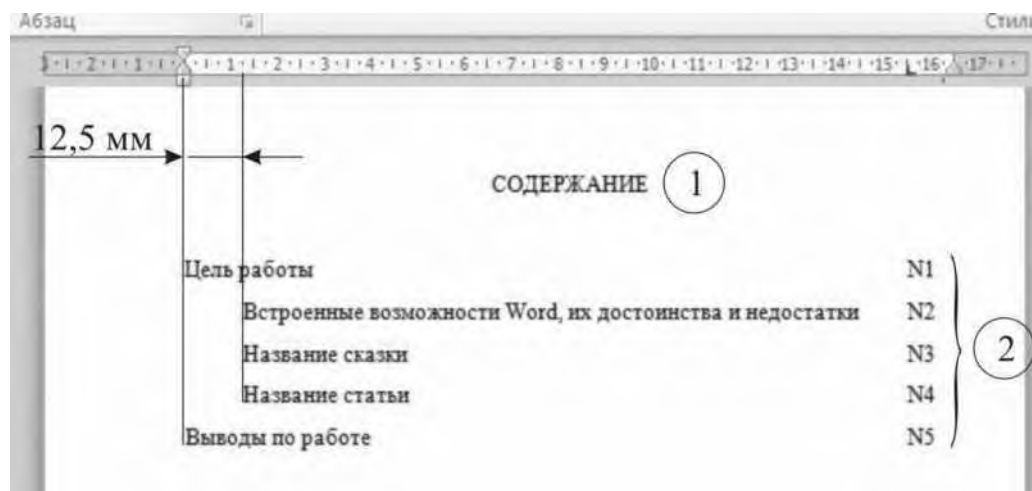
3: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, все прописные. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

4: Шрифт Times New Roman, размер 16 пт, начертание полужирное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

5: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по ширине, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки, позиция табуляции 12 см.

6: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

Оформление Содержания отчета по лабораторной работе



Примечание: в кружках обозначены номера наборов параметров форматирования текста.

1: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, все прописные. Выравнивание по центру, отступы слева 0, справа 0, красной строки нет, интервалы перед 0, после 0, междустрочный 1,5 строки.

2: Шрифт Times New Roman, размер 14 пт, начертание обычное, цвет черный, видоизменений нет. Выравнивание по левому краю, отступы слева 0, справа 0, интервалы перед 0, после 0, междустрочный 1,5 строки.