

GNU Эмулятор ВЫЧИСЛЕНИЯ

- 1. Арифметические команды**
- 2. Сдвиговые команды**
- 3. Логические команды**

Команда сложения:

ADD R1,R2

Схема работы команды:

$R1=R1+R2$

Команда вычитания:

SUB R1,R2

Схема работы команды:

$R1=R1-R2$

Где R_i - Регистр

Пример сложения

Требуется выполнить сложение

$$A = B + C$$

При $B = 78_{10}$ и $C = -34_{10}$

$$A = 44_{10}$$

Псевдо код программы сложения:

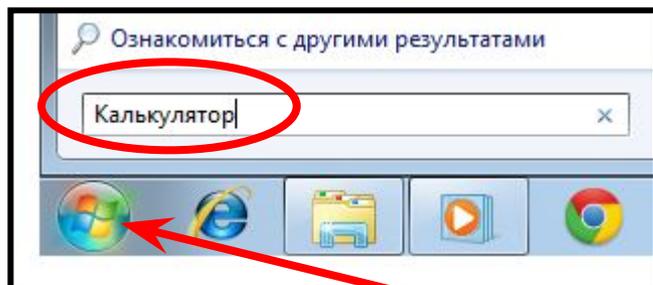
$$AL \leq 78$$

$$BL \leq -34$$

$$AL = AL + BL$$

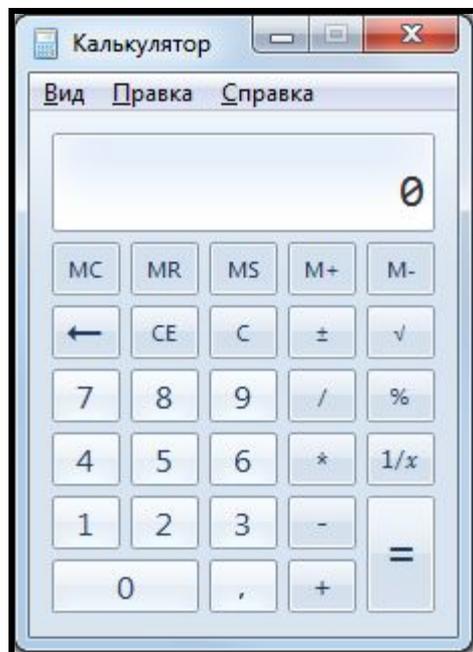
конец

Для эмулятора требуется 16ричная система исчисления !

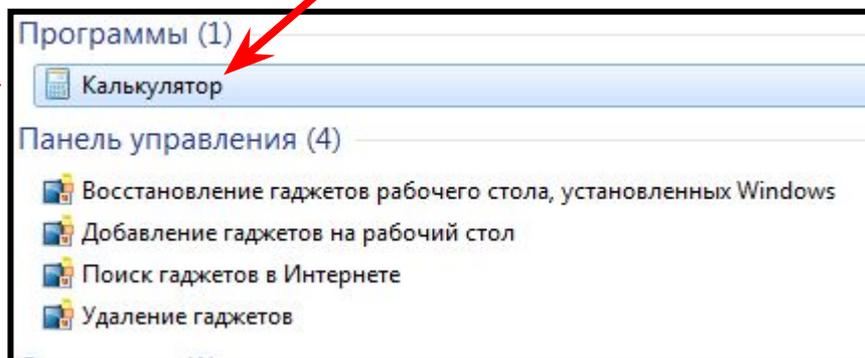


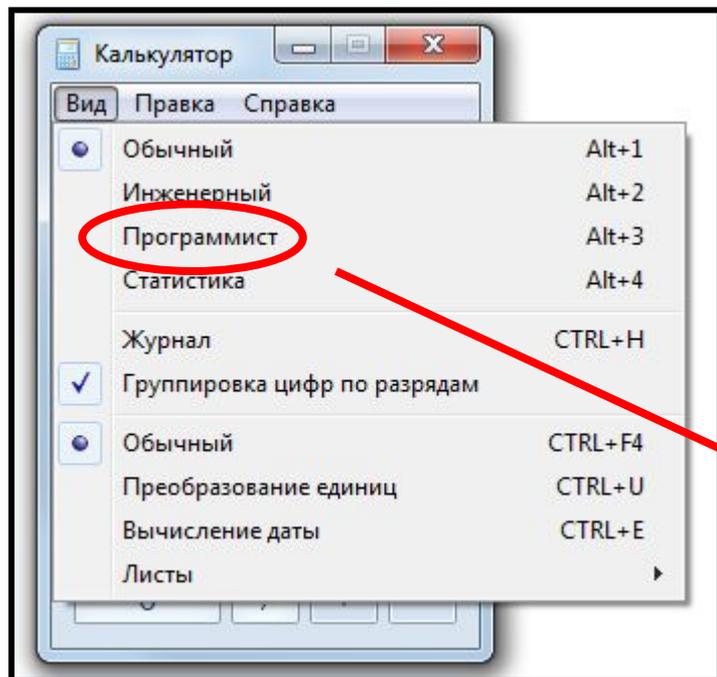
Для получения кодов чисел используем программу «Калькулятор» MS Windows

1. Запустите эмулятор и откройте строку поиска MS Windows, отыщите нужное приложение.



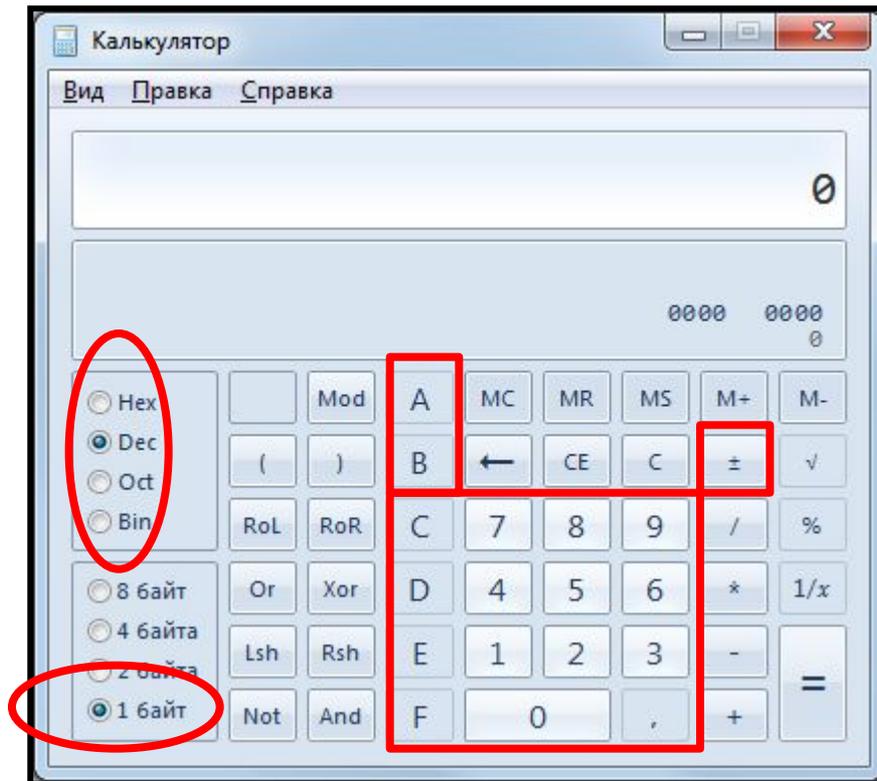
2. Вызовите калькулятор





3. Переведите калькулятор в режим «Программист», используя команду меню «Вид»



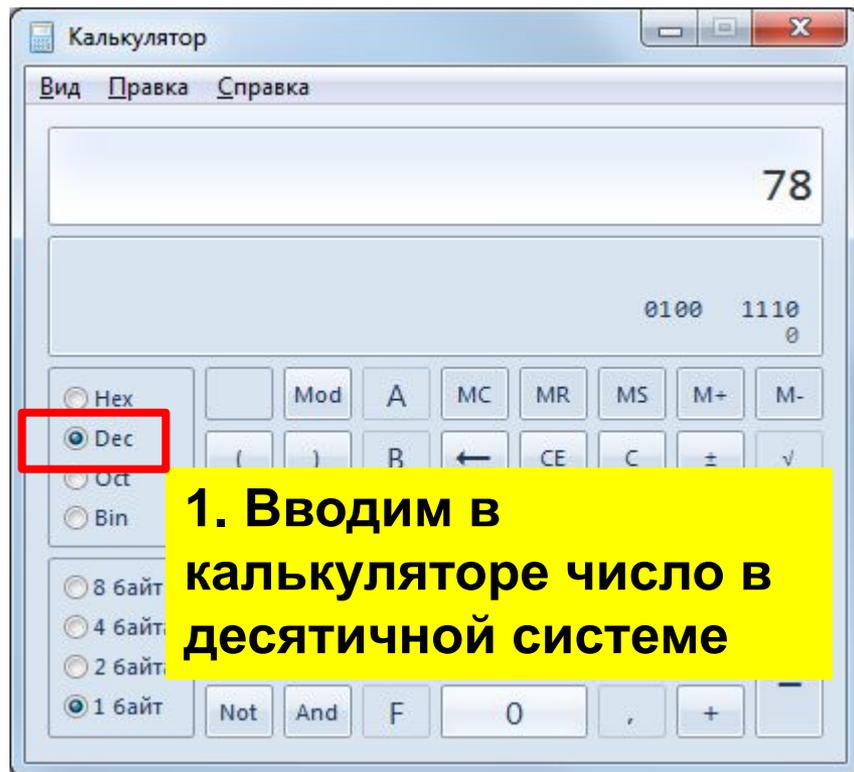


Выберите кодировку символов один байт

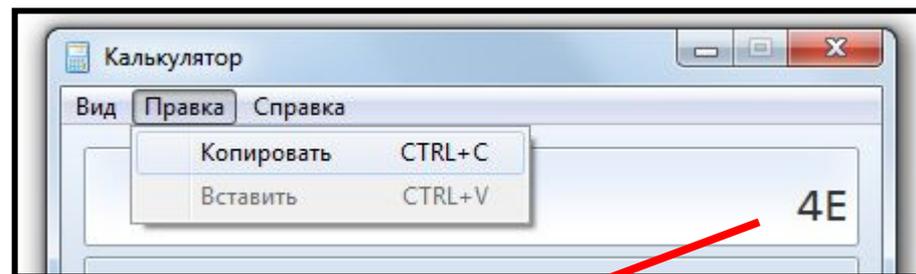
Установите нужную систему исчисления:

- Hex - шестнадцатеричная
- Dec - десятичная
- Oct – восьмеричная
- Bin - двоичная

- Используйте цифровые клавиши для набора нужного кода.
- При смене системы исчисления код будет автоматически переведен в новую систему.
- Смена знака кода \pm



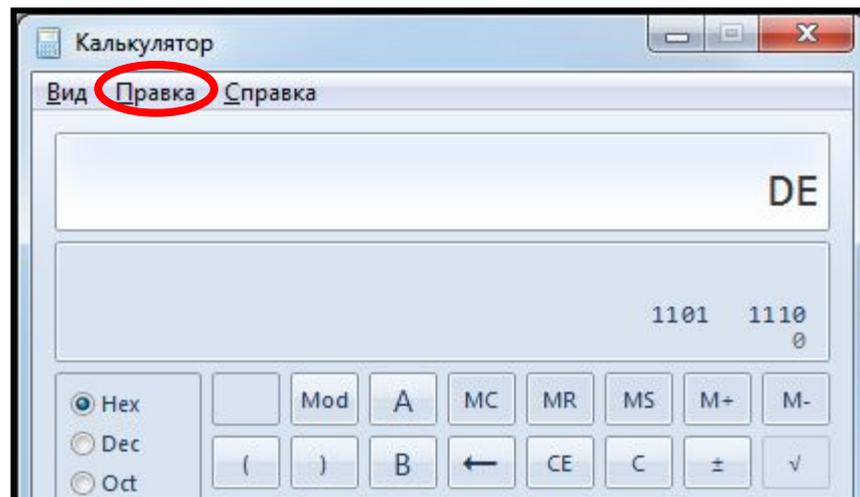
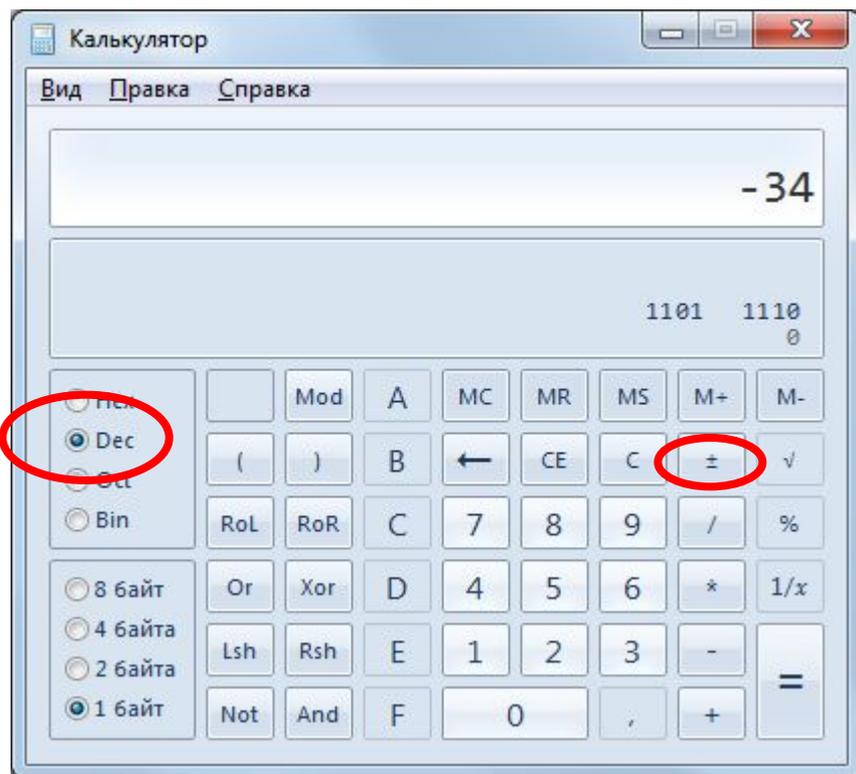
2. Переключаем Dec на Hex режим.
3. Используя команду «Копировать» копируем нужное число и помещаем в редактор эмулятора



```
Source Code  
mov AL, 4E
```

Аналогично поступаем с числом -34
Вводим число в режиме Дес, меняем знак

Переходим в режим Нех и копируем число в эмулятор

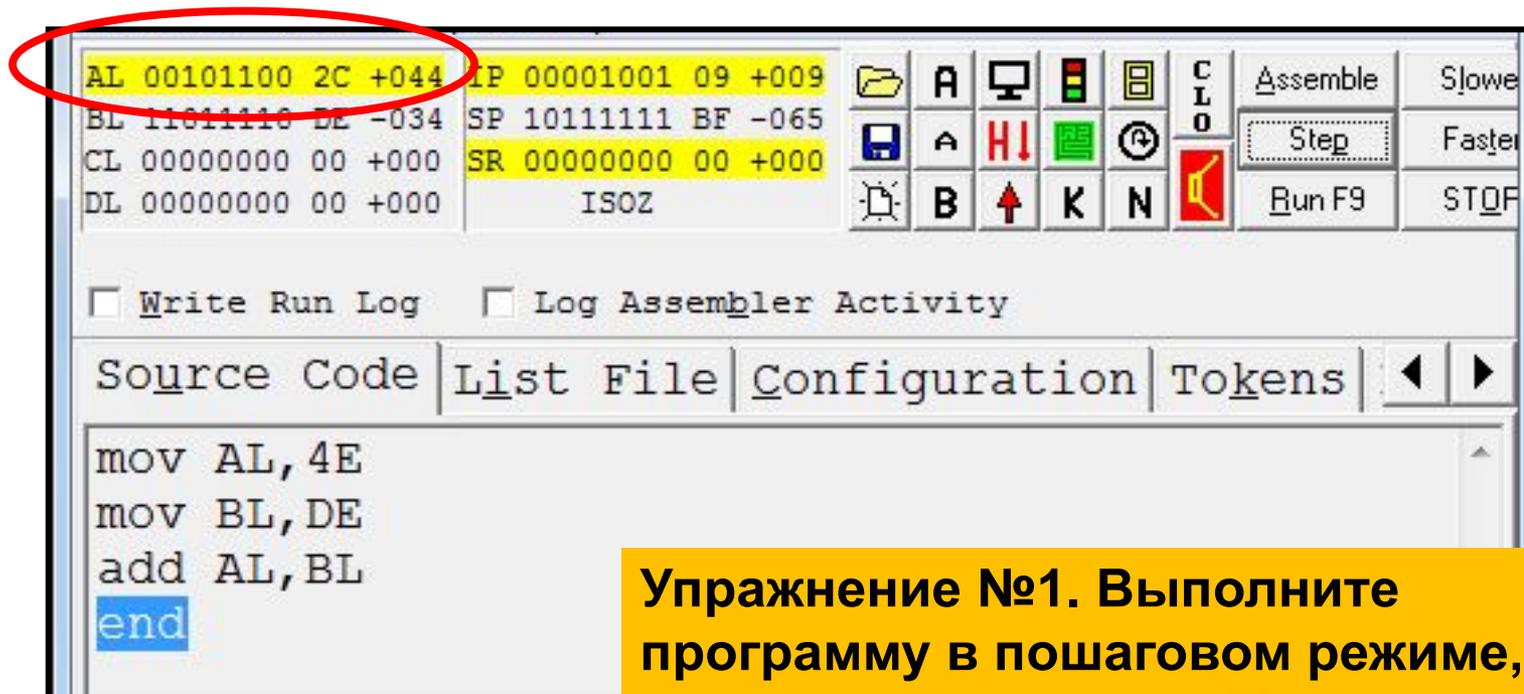


Source Code

```
mov AL, 4E  
mov BL, DE
```

Программа сложения кодов

Сохраните программу в Вашей папке как sum2.asm



Упражнение №1. Выполните программу в пошаговом режиме, результат в регистре AL

- ❑ Диапазон допустимых значений для целых чисел в десятичной системе составит величину -128 до +127.
- ❑ В шестнадцатеричной системе этот диапазон примет значение от 80 до 7F.

Переполнение

- ❑ Для отслеживания состояния переполнения используется бит O(Overflow) регистра SR процессора.
- ❑ До организации перехода в программе при переполнении используются команда JO и JNO.

IP	00000000	00	+000
SP	10111111	BF	-065
SR	00000000	00	+000

ISOZ

JO my_label ;O=1 -Переполнение

JNO my_label ;O=0 – Нет переполнения

Задания

Упражнение №1. Требуется написать программу sumcontrol.asm для вычисления суммы с постоянным числом $10_{10} = 0A_{16}$.

AL <= 0

Выполнять:

AL=AL+0A

если бит 0 = 1 переход Стоп

переход Выполнять

Стоп:

Конец

Контроль знака кода

При выполнении вычитания контроль знака числа производится с помощью бита S(Sign) регистра состояния. Он выставляется в единицу, если было получено отрицательное число при вычитании.

IP	00000000	00	+000
SP	10111111	BF	-065
SR	00000000	00	+000
	ISDZ		

Команды JS и JNS служат для организации переходов при необходимости учета знака кодов – результатов операции.

```
IP 00000000 00 +000  
SP 10111111 BF -065  
SR 00000000 00 +000  
ISDZ
```

Формат команд:

JS my_label ;При S=1

JNS my_label ;При S=0

Упражнение №2.

Контроль отрицательного результата, программа `sigcontrol.asm`. Сравнить два числа в регистрах процессора `CL` и `AL`.

Если код в `CL` меньше чем в `AL` вывести на `VDU` сообщение `YES` иначе вывести сообщение `NO`. Контрольные значения:

`AL=F (15)`

`CL=A (10)`

Псевдо код программы

```
Перейти begin
  “YES”
  “NO”
begin:
AL <= F | 15
CL <= A | 10
Сравнить CL с AL
Если S= 1 перейти на ok
Перейти по
ok:
Вывод “YES”
Перейти fin
по:
Вывод “NO”
fin:
КОНЕЦ
```

Команда умножения:

MUL R1,R2

Схема работы команды:

$R1=R1*R2$

Команда деления:

DIV R1,R2

Схема работы команды:

$R1=R1/R2$

Умножение и деление

- ❑ Операндами команды являются регистры.
- ❑ В качестве второго операнда можно использовать код.
- ❑ Результат деления – целое число, дробная часть усекается !!!

Деление значения в регистре на число 11.
`DIV CL,0B ;CL=CL/0B`

Логические команды

Бит ₁	Бит ₂	AND	OR	XOR
1	1	1	1	0
0	1	0	1	1
1	0	0	1	1
0	0	0	0	0

Схема работы команд:

A1=A1 Command A2

Использование команд:

Использование команд:

Command R,N ;N-код

Command R1,R2; R-регистр

AND A1,A2

OR A1,A2

XOR A1,A2

Контроль активности битов

Упражнение №3. Требуется определить активность 3 бита регистра DL. Маска бита равна $100_2 = 4_{16}$. Выведите контрольные сообщения на VDU:

Find! (найден)

Not find!(не найден)

Программный код проверки активности бита:

Код проверки активности бита:

push DL;регистр в стек

AND DL,4;Маска бита

jz no ;Z=1

jmp next ;Бит установлен

no:

;Бит не установлен

...

next:

pop DL;Извлечь регистр

...

Упражнение №4. Требуется активизировать пятый бит регистра AL. Начальное значение кода в регистре равно нулю. Маска бита $10000_2 = 16_{10} = 10_{16}$. Проверьте значение кода в регистре до установки бита и после. Выведите контрольные сообщения на VDU.

**Команда установки бита:
OR AL,10**

Шифрование кодов и обнуление

Операция XOR по отношению к битам обладает свойством обратимости, что делает ее пригодной для шифрования байтов.

```
;Шифрование  
MOV AL,A ;1010  
MOV BL,2 ;0010 – Ключ  
XOR AL,BL
```

```
;Дешифрование  
XOR AL,BL
```

Обнуление с помощью команды XOR регистр вытекает из ее таблицы истинности.

Сдвиг битов

Сдвиговые команды:

SHL R – Выполнение левого сдвига в регистре.

SHR R – Выполнение правого сдвига в регистре.

Сдвиговые операции позволяют увеличивать или уменьшать в два раза значения в регистре.

Упражнение №6. Требуется получить куб цифры 2. Псевдокод программы:

AL<=01

CL<=03

Выполнять:

левый сдвиг AL

CL=CL-1

**если Z не 1 то Выполнять
конец**

**Контрольное задание №1. Программа
perebor.asm**

**Дана последовательность десятичных чисел:
10,20,30,40,50,1,4,4,100,110.**

- Вычислить сумму чисел до получения переполнения.**
- Сумму поместить в регистр AL**
- При выборе очередного числа из памяти на VDU выводить символ, соответствующий прочитанному коду**
- При переполнении вывести контрольное сообщение ERROR !**

Контрольное задание №2.

Поместить в регистр BL, число 125_{10} . Написать программу `decrement.asm` для последовательного уменьшения значения числа с шагом 2. При получении отрицательного значения вычисления прекратить и вывести контрольное сообщение **STOP !**

Контрольное задание №3. Программа `fact.asm` Вычислить факториал числа 4.
Контрольное значение 24_{10}

Задания

Контрольное задание №4. В регистр CL поместить четырех разрядное двоичное число. Вывести представление числа в символьном виде. Правило отображения:

0 - N

1- Y

Пример. CL=0101

Код: NYNY

Контрольное задание №5. Дана строка «Hello World!» написать программу для шифрования – дешифрования строки по методу XOR кодирования. Ключ код латинской буквы X

Контрольное задание №6. В регистр AL поместить код 0. Используя команду OR последовательно получить числа 5_{10} , 21_{10} , 117_{10} .