МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**Дисципліна «Архітектура комп’ютера.»**

**Лабораторна робота № 7.**  
**Тема: «Додавання(віднімання)чисел у зворотному коді.»**

Виконав: студент гр. 6.1219-2

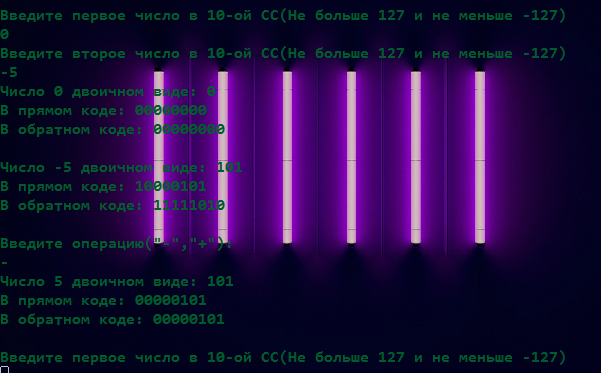
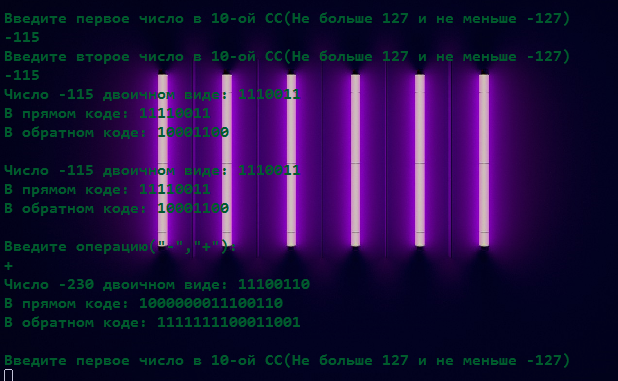
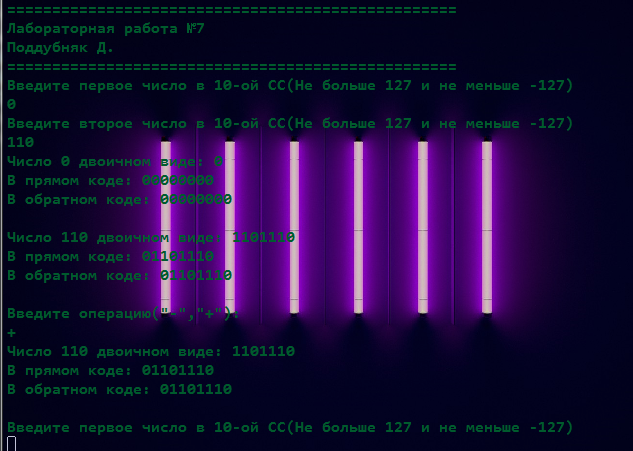
Піддубняк Данііл

Перевірив: Викладач Стреляєв Ю.М.

Запоріжжя

2019

# Скріншоти:



# Код програми:

# Лабораторная работа №7

# Поддубняк Д.

# Функции: binarconverter = перевод из 10 СС в 2 СС, straight\_code = перевод в прямой код, additinally\_code = перевод в дополнительный код, checker = увеличение кол-ва битов если число большое.

# reverse\_code = перевод в обратный код, plus = сложение, minus = вычитание

# Переменные:decimal\_digit = исходное число в 10 СС, straight\_view = число в прямом коде, additinally\_view = число в дополнительном коде

# reverse\_view = число в обратном коде, ispos\_a, ispos\_b = числа в десятичном виде для сравнений

import time

def big\_digit(num, ispositive='+'):

l = len(num) # Длинна числа

if ispositive == '-': # Если отрицательное:

num = '1' + ('0' \* (16 - l - 1)) + num

else: # Если положительное:

num = '0' + ('0' \* (16 - l - 1)) + num

return num

def checker(decimal\_digit):

if decimal\_digit > 127 or decimal\_digit < -127:

decimal\_digit = str(decimal\_digit)

if '-' in decimal\_digit: # Если число отрицательное:

split = decimal\_digit.split('-') # Список для отделения минуса от числа

split[0] = '-' # Отделение минуса

binar\_digit = (binarconverter(split[1]))

straight\_view = big\_digit(binar\_digit, split[0])

reverse\_view = reverse\_code(straight\_view)

print(f'''Число {decimal\_digit} двоичном виде: {binar\_digit}

В прямом коде: {straight\_view}

В обратном коде: {reverse\_view}

''')

return reverse\_view

elif not '-' in decimal\_digit: # Если число положительное:

binar\_digit = binarconverter(decimal\_digit)

straight\_view = big\_digit(binar\_digit)

reverse\_view = straight\_view

print(f'''Число {decimal\_digit} двоичном виде: {binar\_digit}

В прямом коде: {straight\_view}

В обратном коде: {reverse\_view}

''')

return reverse\_view

else:

main(str(decimal\_digit))

def plus(a, b, ispos\_a, ispos\_b): # Принимает числа в обратном коде, числа в десятичной системе для сверок

if int(ispos\_a) > 0 and int(ispos\_b) > 0: # Оба +

a1 = int(a, 2) # Указываем что это число,а не строка

b1 = int(b, 2) # Указываем,что это число, а не строка

c1 = (a1 + b1) & 255 # Сложение чисел с битовым "И"

checker(c1)

elif int(ispos\_a) > 0 and int(ispos\_b) < 0: # Первое + второе -

a1 = int(a, 2) # Указываем что это число,а не строка

b1 = int(b, 2) # Указываем что это число,а не строка

if int(ispos\_a) > ~int(ispos\_b): # Если первое число больше второго(переведенного из отрицательного в пол.)

c1 = (a1 + b1 + 1) & 255 # Сложение чисел с битовым "И"

checker(c1)

else:

c1 = ~(255 - (a1 + b1+1) & 255) # Сложение чисел с битовым "И",инверсией знака

checker(c1)

elif int(ispos\_a) < 0 and int(ispos\_b) > 0: # Первое - второе +

a1 = int(a, 2)

b1 = int(b, 2)

c1 = (a1 + b1 + 1) & 255

checker(c1)

elif int(ispos\_a) < 0 and int(ispos\_b) < 0: # Оба -

a1 = int(a, 2)

b1 = int(b, 2)

c1 = ~(510 - (a1 + b1 + 1)) # Сложение чисел с инверсией знака

checker(c1)

elif int(ispos\_a) == 0 and int(ispos\_b) < 0:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = ~(255 -(a1 + b1 + 1) & 255) # Сложение чисел с инверсией знака

checker(c1)

elif int(ispos\_b) == 0 and int(ispos\_a) < 0:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = ~(255 -(a1 + b1 + 1) & 255) # Сложение чисел с инверсией знака

checker(c1)

else:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = (a1+b1)

checker(c1)

def minus(a, b, ispos\_a, ispos\_b): # Принимает числа в обратном коде, числа в десятичной системе для сверок

if int(ispos\_a) > 0 and int(ispos\_b) > 0: # Оба +

a1 = int(a, 2)

b1 = int(b, 2)

if int(ispos\_b) > int(ispos\_a): # Если первое число больше второго

c1 = (a1 - b1)

checker(c1)

else:

c1 = (a1 - b1) & 255 # Иначе - вычитание с побитовым "И"

checker(c1)

elif int(ispos\_a) > 0 and int(ispos\_b) < 0: # Первое + второе -

a1 = int(a, 2)

b1 = int(b, 2)

if int(ispos\_a) > ~int(ispos\_b): # Если первое число больше второго(с инверсией знака)

c1 = (a1 - b1 - 1) & 255 # вычитание с побитовым "И"

checker(c1)

else:

c1 = (a1 - b1 - 1) & 255 # вычитание с побитовым "И"

checker(c1)

elif int(ispos\_a) < 0 and int(ispos\_b) > 0: # Первое - второе +

a1 = int(a, 2)

b1 = int(b, 2)

if int(ispos\_b) > ~(int(ispos\_a)): # Если первое больше второго(с инверсией знака)

c1 = ~(255 -((a1 - b1 + 1) & 255)) # вычитание с побитовым "И", инверсией знака

checker(c1)

else:

c1 = (a1 - b1 - 1) & 255 # вычитание с побитовым "И"

checker(c1)

elif int(ispos\_a) < 0 and int(ispos\_b) < 0: # Оба -

a1 = int(a, 2)

b1 = int(b, 2)

c1 = (a1 - b1)

checker(c1)

elif int(ispos\_a) == 0 and int(ispos\_b) < 0:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = 255 -(a1+b1) & 255 # Сложение чисел с инверсией знака

checker(c1)

elif int(ispos\_b) == 0 and int(ispos\_a) < 0:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = ~(255 -(a1 - b1 + 1) & 255) # Сложение чисел с инверсией знака

checker(c1)

else:

a1 = int(a, 2)

b1 = int(b, 2)

c1 = (a1 - b1)

checker(c1)

def main(decimal\_digit):

if '-' in decimal\_digit: # Если число отрицательное:

split = decimal\_digit.split('-') # Список для отделения минуса от числа

split[0] = '-' # Отделение минуса

binar\_digit = (binarconverter(split[1]))

straight\_view = straight\_code(binar\_digit, split[0])

reverse\_view = reverse\_code(straight\_view)

print(f'''Число {decimal\_digit} двоичном виде: {binar\_digit}

В прямом коде: {straight\_view}

В обратном коде: {reverse\_view}

''')

return reverse\_view

elif not '-' in decimal\_digit: # Если число положительное:

binar\_digit = binarconverter(decimal\_digit)

straight\_view = straight\_code(binar\_digit)

reverse\_view = straight\_view

print(f'''Число {decimal\_digit} двоичном виде: {binar\_digit}

В прямом коде: {straight\_view}

В обратном коде: {reverse\_view}

''')

return reverse\_view

def binarconverter(x, res=''): # перевод десятичного числа в двоичную СС.

x = int(x)

if x == 0:

return '0'

else:

while x > 0:

res = ('0' if x % 2 == 0 else '1') + res # Формула перевода в бинарный вид

x //= 2

return str(res)

def straight\_code(num, ispositive='+'):

l = len(num) # Длинна числа

if ispositive == '-': # Если отрицательное:

num = '1' + ('0' \* (8 - l - 1)) + num

else: # Если положительное:

num = '0' + ('0' \* (8 - l - 1)) + num

return num

def reverse\_code(num):

new\_num = '' # Переменная для коррекции

num\_part = ''

if num[0] == '0': # Инверсия путем добавления в переменную для коррекции итоговых значений

num\_part = num[0]

num = num[1:]

elif num[0] == '1':

num\_part = num[0]

num = num[1:]

for i in range(len(num)):

if num[i] == '0':

new\_num += '1'

elif num[i] == '1':

new\_num += '0'

return num\_part + new\_num # Соединение итогового числа с разрядной цифрой

print("=" \* 50)

print('''Лабораторная работа №7

Поддубняк Д.''')

print("=" \* 50)

while True:

first\_digit = input('Введите первое число в 10-ой СС(Не больше 127 и не меньше -127)\n')

second\_digit = input('Введите второе число в 10-ой СС(Не больше 127 и не меньше -127)\n')

try:

if int(first\_digit) <= 127 and int(first\_digit) >= -127: # Ограничение числа(Конфигурабельно)

try:

converted\_f\_digit = main(first\_digit)

converted\_s\_digit = main(second\_digit)

except Exception:

print("Неверно введено число.")

continue

todo = input('Введите операцию("-","+"):\n')

if todo == '+':

plus(converted\_f\_digit, converted\_s\_digit, first\_digit, second\_digit)

time.sleep(2)

continue

elif todo == '-':

minus(converted\_f\_digit, converted\_s\_digit, first\_digit, second\_digit)

time.sleep(2)

continue

else:

print("Неверно выбрана операция.")

continue

else:

print("Система рассчитана на число меньше 128|болше -128")

continue

except Exception:

print("Ошибка ввода.")

continue