Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп’ютерних систем

Лабораторна робота №2

з курсу «Комп’ютерні системи»
студента 3 курсу
групи СА-КІ
Колоскова Нікіти

Київ
2019

Варіанти завдань: a c a (майже халява, лол)

Для тестування роботи програм було використані випадкові значеннння, отримані з генераторів випадкових чисел з модуля random у Python. У звіті наведені повний вивід програми на прикладі одного запуску та результати програми для множинних запусків.

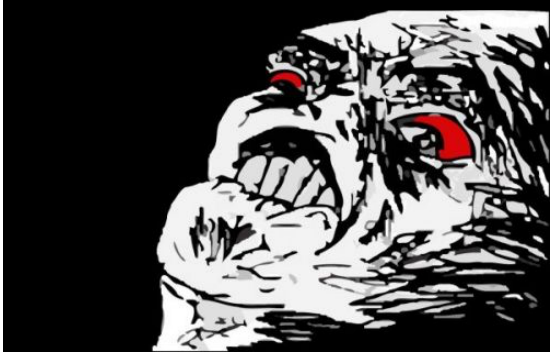
1. Множення двійкових чисел як є

```
nikita@MacBook-Air-Apple-3:~/University (Labs)/semester 6/computer_systems/lab2$ python test.py
Multiplication check
a1 =  7
b1 =  19
Multiplicand: 00000000000000000000000000000111
Multiplier: 00000000000000000000000000010011
Initializing register: 0000000000000000000000000000000000000000000000000000000000000000

Multiplier LSB=1: adding multiplicand to product
Product: 00000000000000000000000000000000000000000000000000000000000000111
Shifting multiplicand to left: 00000000000000000000000000001110
Shifting multiplier to right: 00000000000000000000000000001001

Multiplier LSB=1: adding multiplicand to product
Product: 00000000000000000000000000000000000000000000000000000000000010101
Shifting multiplicand to left: 00000000000000000000000000011100
Shifting multiplier to right: 00000000000000000000000000000100

Shifting multiplicand to left: 00000000000000000000000000111000
Shifting multiplier to right: 00000000000000000000000000000010

Shifting multiplicand to left: 00000000000000000000000001110000
Shifting multiplier to right: 00000000000000000000000000000001

Multiplier LSB=1: adding multiplicand to product
Product: 00000000000000000000000000000000000000000000000000000010000101
Shifting multiplicand to left: 00000000000000000000000011100000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000000000111000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000000001110000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000000011100000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000000111000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000001110000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000011100000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000111000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000001110000000000000
Shifting multiplier to right: 00000000000000000000000000000000
```

```
Shifting multiplicand to left: 00000000000000011100000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000111000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000001110000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000011100000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000111000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000001110000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000011100000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000111000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000001110000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000011100000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000111000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00001110000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00011100000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00111000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 01110000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 11100000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 11000000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000
```

```
Shifting multiplicand to left: 10000000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Shifting multiplicand to left: 00000000000000000000000000000000
Shifting multiplier to right: 00000000000000000000000000000000

Result:  133
Expected result:  133
```

```
Multiplication check
a1 =  18
b1 =  37
Result:  666
Expected result:  666
```



```
Multiplication check
a1 =  48
b1 =  28
Result:  1344
Expected result:  1344
```

2. Ділення двійкових чисел: частка та залишок в одному регістрі

```
Division check
a2 =  70
b2 =  25
Dividend: 00000000000000000000000001000110
Divisor:  11001000000000000000000000000000
Initial register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  01100100000000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00110010000000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00011001000000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00001100100000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000110010000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000011001000000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000001100100000000000000000000
Register: 00000000000000000000000001000110000000000000000000000000000000
```

```
Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000110010000000000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000011001000000000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000001100100000000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000110010000000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000011001000000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000001100100000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000110010000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000011001000000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0
```

```
Divisor:  00000000000000001100100000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000110010000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000011001000000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000001100100000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000110010000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000011001000000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000001100100000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000000110010000
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000000011001000
Register: 00000000000000000000000010001100000000000000000000000000000000
```

```
Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000000001100100
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000000000110010
Register: 00000000000000000000000010001100000000000000000000000000000000

Divisor is less then dividend
Last bit in quotient is 1
Shifting register to left
Divisor:  00000000000000000000000000011001
Register: 00000000000000000000000000101000000000000000000000000000000001

Divisor is greater then dividend
The last quotient bit is set to 0

Divisor:  00000000000000000000000000001100
Register: 00000000000000000000000000010100000000000000000000000000000010

Result:  2
Expected result:  2
```

```
Division check
a2 =  88
b2 =  2

Result:  44
Expected result:  44
```

```
Division check
a2 =  94
b2 =  21

Result:  4
Expected result:  4
```

3. АЯЯЯЙ 754: додавання

```
Floating point numbers check
a3 =  5.105868013129237
b3 =  1.0857713645152471
floating a3:  01000000101000110110001101000101
floating b3:  00111111100010101111101010001110
Exponents difference:  2
Shifted mantissa of least number:  010001010111111010100011
Summed mantissa:  10001100010000111101000
Addition result:  01000000110001100010000111101000
Result:  6.191638946533203
Expected result:  6.191639377644485

Floating point numbers check
a3 =  2.4087058994732793
b3 =  4.336843392618862
floating a3:  01000000000110100010100000111101
floating b3:  01000001000101011000111011011100
Exponents difference:  1
Shifted mantissa of least number:  10011010001010000011110
Summed mantissa:  10101111101101110001010
Addition result:  01000000110101111101101110001010
Result:  6.745549201965332
Expected result:  6.745549292092141

Floating point numbers check
a3 =  3.093576742045764
b3 =  0.26797241752600987
floating a3:  01000000100010111111110100101001
floating b3:  00111110100010010011001110101110
Exponents difference:  3
Shifted mantissa of least number:  00100010010011001110101
Summed mantissa:  10101110010001110011110
Addition result:  01000000010101110010001110011110
Result:  3.361548900604248
Expected result:  3.3615491595717737
```

Висновки: лабораторні роботи потрібно робити завчасно. О третій годині ночі імпакту з них майже жодного не залишається :(