МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

АЛГОРИТМЫ АЛГЕБРЫ И ТЕОРИИ ЧИСЕЛ

ЛАБОРАТОРНАЯ РАБОТА №1

студента 4 курса 431 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Никитина Арсения Владимировича

Проверил
доцент _____ А. С. Гераськин

Саратов 2022

СОДЕРЖАНИЕ

1 Задание лабораторной работы

Решите сравнение вида $ax \equiv b \ (mod \ m)$ с помощью алгоритма Евклида.

2 Теоретическая часть

Если НОД(a, m) чисел a и m равен d и d делит b, то сравнение $ax \equiv b \ (mod \ m)$ имеет d решений. Если же d не делит b, то сравнение не имеет решений.

После нахождения d и выполнения условий, требуется разделить соотношение на это число d.

Пусть после разделения соотношения оно имеет вид:

$$a_1 x \equiv b_1 \ (mod \, m_1)$$

Теперь имеем $\mathbf{GCD}(a, m) = 1$, поэтому можно воспользоваться расширенным алгоритмом Евклида и получить коэффициенты разложения Безу:

$$1 = a_1 q + m_1 q \Rightarrow b_1 = a_1 b_1 q + m_1 r b_1 \Rightarrow a_1 b_1 q - b_1 = -m_1 r b_1$$

.

Затем все решения сравнения можно найти по формуле:

$$x \equiv x_0 + m_1 k \ (mod \, m), \ k = \overline{0, d}$$

3 Практическая часть

3.1 Пример работы алгоритма

```
Ввести числа a, b, m - \enter
Выход из программы - 2
Введите значение:
Введите значение a: 12
Введите значение b: 9
Введите значение m: 21

Сравнение имеет вид 12 * x ≡ 9 (mod 21)
НОД(12, 21) = 3
И 9 делится на 3. Это означает что сравнение имеет 3 решений.
Разделим сравнение и его модуль на 3 и получим: 4 * x ≡ 3 (mod 7)
x_0= 6
x_1=13
x_2=20

Ввести числа a, b, m - \enter
Выход из программы - 2
Введите значение:
Введите значение a: 13
Введите значение b: 26
Введите значение m: 169

Сравнение имеет вид 13 * x ≡ 26 (mod 169)
НОД(13, 169) = 13
И 26 делится на 13. Это означает что сравнение имеет 13 решений.
Разделим сравнение и его модуль на 13 и получим: 1 * x ≡ 2 (mod 13)
x_0= 2
x_1=15
x_2=28
x_3=41
x_4=54
x_5=67
x_6=80
x_7=93
x_8=106
x_9=119
x_10=132
x_11=145
x_12=158

Ввести числа a, b, m - \enter
Выход из программы - 2
Введите значение: 2
Работа программы завершена
```

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
1  def bezout_recursive(a, b):
2
3      if not b:
4          return (1, 0, a)
5      y, x, g = bezout_recursive(b, a % b)
6      return (x, y - (a // b) * x, g)
7
8
9  def gcd(a, b):
10     if b == 0:
11         return a
```

```
12      else:
13          return gcd(b, a % b)
14
15
16  def main():
17
18      while True:
19
20          print('\nВвести числа a, b, m - \enter')
21          print('Выход из программы - 2')
22
23          try:
24              value = int(input('Введите значение: '))
25          except ValueError:
26              value = 1
27
28          if value == 1:
29
30              a = int(input('Введите значение a: '))
31              b = int(input('Введите значение b: '))
32              m = int(input('Введите значение m: '))
33
34              a %= m
35              b %= m
36
37              print(f'\nСравнение имеет вид {a} * x \u2261 {b} (mod {m})')
38
39              g = gcd(a,m)
40              print(f'НОД({a}, {m}) = {g}')
41
42              if not b % g:
43
44                  print(f'И {b} делится на {g}. Это означает, что всего
                    ↪ решений'\
45                      f' сравнения: {g}.')
46
47                  a //= g
48                  b //= g
49                  m_save = m
50                  m //= g
51
```

```
52                  print(f'Разделим сравнение и его модуль на {g} и получим:'\
53                      f' {a} * x \u2261 {b} (mod {m})')
54
55                  q, _, _ = bezout_recursive(a, m)
56
57                  x_0 = b * q % m_save
58
59                  print('Решения сравнения:')
60
61                  print('x_0 =', x_0)
62
63                  for i in range(1, g):
64                      print(f'x_{i} = {(x_0 + i * m) % m_save}')
65
66              else:
67                  print(f'Сравнение не имеет решений, так как {b} не делится на
                    ↪ {g}.')
68
69
70          if value == 2:
71              print('Работа программы завершена')
72              return
73
74
75  if __name__ == "__main__":
76      main()
```