HomeTask03

Выполнил: Миронов Александр БПИ194

Ошибка 1

1.

public int gcd(int x, int y)  
{  
 int t;  
   
 if(x < 0) x = -x;  
 if(y < 0) y = -y;  
   
 while(y != 0)  
 {  
 if(y > x)   
 {   
 x = y-x;  
 y = y-x;  
 x = x+y;  
 }  
   
 if(y == 0) return x;  
   
 t = y;  
 y = x%y;  
 x = t;  
 }  
 return x;  
}

2. gcd(-2147483648,- 2147483648)

3. Ожидается 2147483648, получен ответ -2147483648

4.

public long gcd(int x, int y)  
{  
 int t;  
 if (x==Integer.*MIN\_VALUE* && x==y)  
 return -(long)Integer.*MIN\_VALUE*;  
 if(x < 0) x = -x;  
 if(y < 0) y = -y;  
  
 while(y != 0)  
 {  
 if(y > x)  
 {  
 x = y-x;  
 y = y-x;  
 x = x+y;  
 }  
  
 if(y == 0) return x;  
  
 t = y;  
 y = x%y;  
 x = t;  
 }  
 return x;  
}

Ошибка 2

1.

public int gcd(int x, int y)  
{  
 int t;  
 if (x==Integer.*MIN\_VALUE* && x==y)  
 return -(long)Integer.*MIN\_VALUE*;  
 if(x < 0) x = -x;  
 if(y < 0) y = -y;  
   
 while(y != 0)  
 {  
 if(y > x)   
 {   
 x = y-x;  
 y = y-x;  
 x = x+y;  
 }  
   
 if(y == 0) return x;  
   
 t = y;  
 y = x%y;  
 x = t;  
 }  
 return x;  
}

2. gcd(-2147483648,2), gcd(-2147483648,4) и другие делители числа 2147483648 в качестве второго аргумента

3. Ожидается 2, 4… ~~Получен ответ~~, программа уходит в бесконечный цикл, так как не может перевести -2147483648 в положительный эквивалент

4.

public int gcd(int x, int y)  
{  
 int t;  
 if (x==Integer.*MIN\_VALUE* && x==y)  
 return -(long)Integer.*MIN\_VALUE*;  
 if (x==Integer.*MIN\_VALUE*)  
 x=1073741824;  
 if (y==Integer.*MIN\_VALUE*)  
 y=1073741824;  
  
 if(x < 0) x = -x;  
 if(y < 0) y = -y;  
  
 while(y != 0)  
 {  
 if(y > x)  
 {  
 x = y-x;  
 y = y-x;  
 x = x+y;  
 }  
  
 if(y == 0) return x;  
  
 t = y;  
 y = x%y;  
 x = t;  
 }  
 return x;  
}

P.S. На счёт исправления ошибки 2: поскольку Integer.MIN\_VALUE это степень двойки, то понижая степень аргумента равного Integer.MIN\_VALUE НОД в результате будет тем же, если второй аргумент сам не принимает значение Integer.MIN\_VALUE, однако этот случай исключён после исправления ошибки 1.