

1. Техническое задание

Реализовать нахождение наибольшего общего делителя (НОД) для массива чисел при помощи EDSAC.

2. Метод решения

Идея алгоритма заключается в том, что мы вычитаем из большего числа меньшее и заменяем первое на их разность до тех пор, пока их разность не станет равна нулю. В таком случае уменьшаемое и вычитаемое как раз и будут искомым числом.

Method solution

The highest number that is found in the list is being subtracted from each and every number until a value is found that when subtracted from every number in the list will result in zero. Example:

- $A = 10; B = 2$
 - $A - B = 8$
 - $A - B = 6$
 - $A - B = 4$
 - $A - B = 2$
 - $A - B = 0$
- Ответ = 2

3. Программа Orders 1

```
Working_orders 1
31 T103S
32 X0S
33 T0S [acc = 0]
34 A98S [acc = lineOffFirst]
35 T99S [index = lineOffFirst; acc = 0]
36 A100S [acc = first]
37 T94S [bigNum = acc; acc = 0]
38 A97S [acc = size]
39 S90S [acc -= 1; nextElement1]
40 T92S [count = size; acc = 0]
41 T0S [acc = 0]
42 A99S [acc = index]
43 L0L [(var index)]
44 A96S [A(var index)S]
45 T46S [put fun 1]
46 A1S [result first]
47 T95S [numCheck = acc; acc = 0]
48 A94S [acc += bigNum]
49 S95S [acc -= numCheck]
50 G53S [goHere]
51 A95S [acc += NumCheck]
52 T94S [bigNum = acc; acc = 0]
53 A99S [goHere]
54 A90S [index += 1]
55 T99S [var index = acc; acc = 0]
56 A92S [acc = count; acc = 0]
57 E39S [nextElement1]
58 A94S [acc = bigNum]
59 A90S [bigNum += 1]
60 T91S [var one = bigNum]
61 T0S [acc = 0; restart]
62 A98S [acc = lineOffFirst]
63 T99S [index = lineOffFirst; acc = 0]
64 A97S [acc = size]
65 S90S [acc -= 1]
66 T92S [count = size; acc = 0]
67 A91S [acc += var one]
68 S90S [acc -= 1]
69 T91S [var one += 1; acc = 0]
```

The Edsac Simulator

File Edit Library Edsac Window Help

Initial Orders 1 ☐ Stop Bell ☐ Real Time ☒ Sound ☒ Hints ☒ Short Tanks

WORD 91 Order = P 2 L Integer 91S = 5 Fraction 90L = 0.00007629400

Edsac

Output From: Working_orders 1

Clear Reset Start Stop Single E.P.

SCT Order Tank Multiplier Long Tank Multipliland Acc

Dial

```
Working_orders 1
69 T91S [var one += 1; acc = 0]
70 G103S [nextElement]
71 T0S [acc = 0]
72 A99S [acc = index]
73 L0L [(var index)]
74 A96S [A(var index)S]
75 T76S [put fun 1]
76 A1S [result first]
77 S91S [subtract; acc =0-10]
78 E77S [acc -= var one; acc = -10]
79 A91S [check acc += var one; acc = 0]
80 S90S [acc -= 1; acc = -1]
81 E61S [acc == 0; acc = -1]
82 T0S [acc = 0]
83 A90S [acc += 1]
84 A99S [acc += var index]
85 T99S [var index = acc; acc = 0]
86 A92S [acc = count]
87 S90S [acc += 1]
88 U92S [count = acc ;acc = count]
89 E70S [nextElement]
90 P0L [val one = 1]
91 P0L [var one = 1]
92 P0S [var count = 0]
93 P0S [var answer = 0]
94 P0S [var bigNum = 0]
95 P0S [var numCheck = 0]
96 A0S [fun 1]
97 P1L [size ;3]
98 P50S [listStart]
99 P0S [var index]
100 [array] P5S [first] [2*3 + 0 = 6]
101 P2L [second] [2*3 + 1 = 7]
102 P7L [third] [2*3 + 0 = 8]
103 [End]
104
105
```

The Edsac Simulator

File Edit Library Edsac Window Help

Initial Orders 1 ☐ Stop Bell ☐ Real Time ☒ Sound ☒ Hints ☒ Short Tanks

WORD 91 Order = P 2 L Integer 91S = 5 Fraction 90L = 0.00007629400

Edsac

Output From: Working_orders 1

Clear Reset Start Stop Single E.P.

SCT Order Tank Multiplier Long Tank Multipliland Acc

Dial

[Реализовать нахождение наибольшего общего делителя (НОД) для массива чисел при помощи EDSAC; Ответ сохраняется в строке 91]

```
[31] T103S
[32] X0S
[33] T0S      [acc = 0]
[34] A98S      [acc = lineOfFirst]
[35] T99S      [index = lineOfFirst; acc = 0]
[36] A100S     [acc = first]
[37] T94S      [bigNum = acc; acc = 0]
[38] A97S      [acc = size]
[39] S90S      [acc -= 1; nextElement1]
[40] T92S      [count = size; acc = 0]
[41] T0S      [acc = 0]
[42] A99S      [acc = index]
[43] L0L      [(var index)]
[44] A96S      [A(var index)S]
[45] T46S      [put fun 1]
[46] A1S      [result first]
[47] T95S      [numCheck = acc; acc = 0]
[48] A94S      [acc += bigNum]
[49] S95S      [acc -= numCheck]
[50] G53S      [goHere]
[51] A95S      [acc += NumCheck]
[52] T94S      [bigNum = acc; acc = 0]
[53] A99S      [goHere]
[54] A90S      [index += 1]
[55] T99S      [var index = acc; acc = 0]
[56] A92S      [acc = count; acc = 0]
[57] E39S      [nextElement1]
```

Начиная со строки 31 и заканчивая строкой 57, этот алгоритм представляет собой цикл, предназначенный для поиска наибольшего числа в списке. Он начинается с добавления первого числа в список, а затем, начиная со второго числа в списке, вычитает его. Если результат вычитания в накопителе положительный, то текущее число в списке становится самым большим.

[58] A94S	[acc = bigNum]
[59] A90S	[bigNum += 1]
[60] T91S	[var one = bigNum]
[61] T0S	[acc = 0; restart]

Добавляет 1 к самому большому найденному числу. Основной код, начинающийся со строки 62, должен быть вычитаем на 1 в процессе настройки, и именно поэтому мы добавляем 1, чтобы не потерять это число.

[62] A98S	[acc = lineOfFirst]
[63] T99S	[index = lineOfFirst; acc = 0]
[64] A97S	[acc = size]
[65] S90S	[acc -= 1]
[66] T92S	[count = size; acc = 0]
[67] A91S	[acc += var one]
[68] S90S	[acc -= 1]
[69] T91S	[acc = 0]

Этот код запускает цикл. Сначала он добавляет строку первого элемента в индекс, чтобы можно было вносить изменения. После этого он уменьшает размер на единицу, потому что ноль считается положительным значением. После этого мы вычитаем единицу из самого большого числа, которое мы нашли.

[70] G103S	[nextElement]
[71] T0S	[acc = 0]
[72] A99S	[acc = index]
[73] L0L	[(var index)]
[74] A96S	[A(var index)S]
[75] T76S	[put fun 1]
[76] A1S	[result first]
[77] S91S	[subtract; acc = 0-10]
[78] E77S	[acc -= var one; acc = -10]
[79] A91S	[check acc += var one; acc = 0]
[80] S90S	[acc -= 1; acc = -1]
[81] E61S	[acc == 0; acc = -1]

Это цикл, который проверяет, равно ли значение, полученное путем вычитания, нулю, потому что деление - это то же самое, что и вычитание. Если в строке 81 значение в накопителе положительное, то это означает, что число не делится, что приводит к перезапуску алгоритма и уменьшению наибольшего значения.

[82] T0S	[acc = 0]
[83] A90S	[acc += 1]
[84] A99S	[acc += var index]

```

[85] T99S      [var index = acc; acc = 0]
[86] A92S      [acc = count]
[87] S90S      [acc += 1]
[88] U92S      [count = acc ;acc = count]
[89] E70S      [nextElement]

```

Если значение равно нулю, то цикл переходит к следующему значению в списке и проверяет его. Этот процесс повторяется для каждого элемента в списке. Если все значения будут разделены на ноль, то счетчик станет равным нулю и остановит

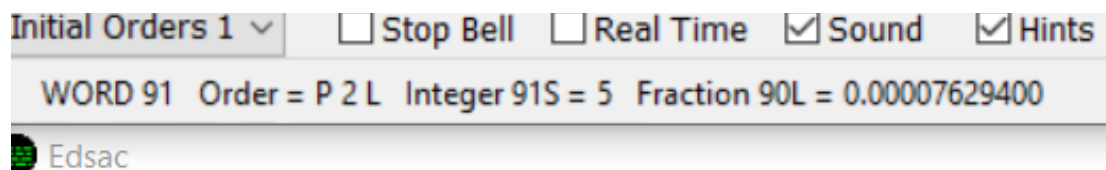
```

[90] P0L      [val one = 1]
[91] P0L      [var one = 1]
[92] P0S      [var count = 0]
[93] P0S      [var answer = 0]
[94] P0S      [var bigNum = 0]
[95] P0S      [var numCheck = 0]
[96] A0S      [fun 1]
[97] P1L      [size ;3]
[98] P50S     [listStart]
[99] P0S      [var index]
[100] [array] P5S [first]  [2*3 + 0 = 6]
[101]         P2L [second] [2*3 + 1 = 7]
[102]         P7L [third]  [2*3 + 0 = 8]
[103] [End]

```

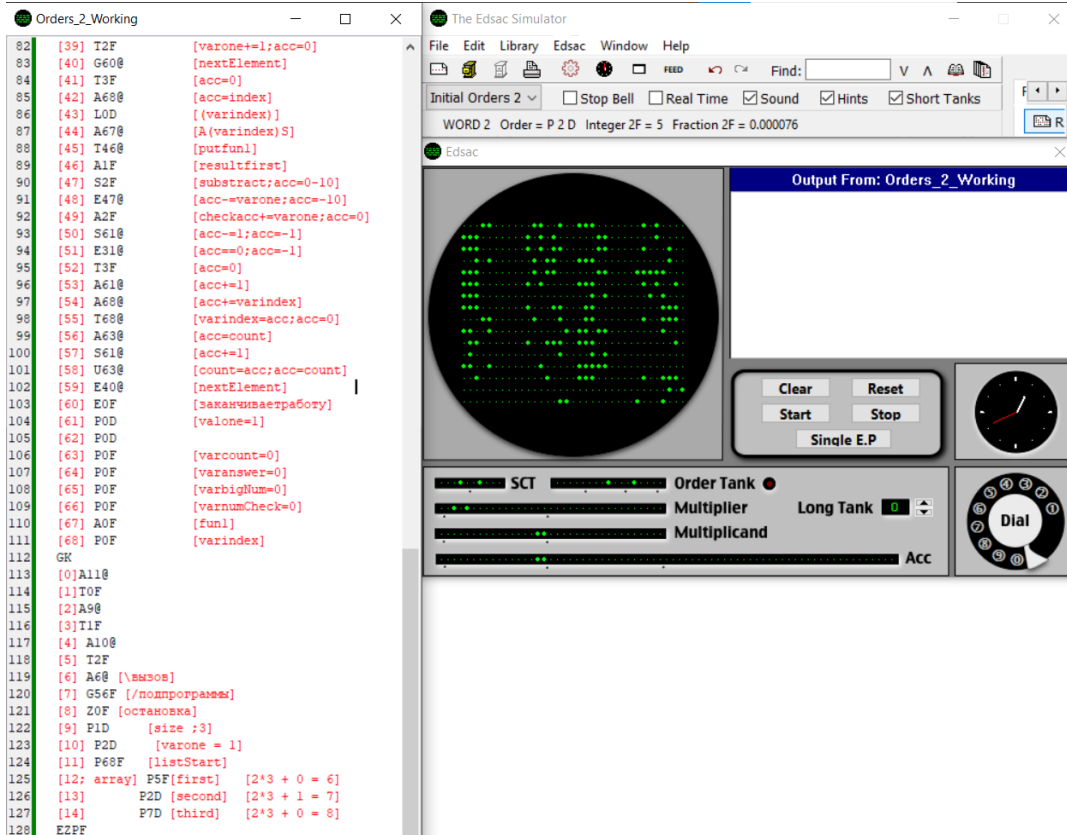
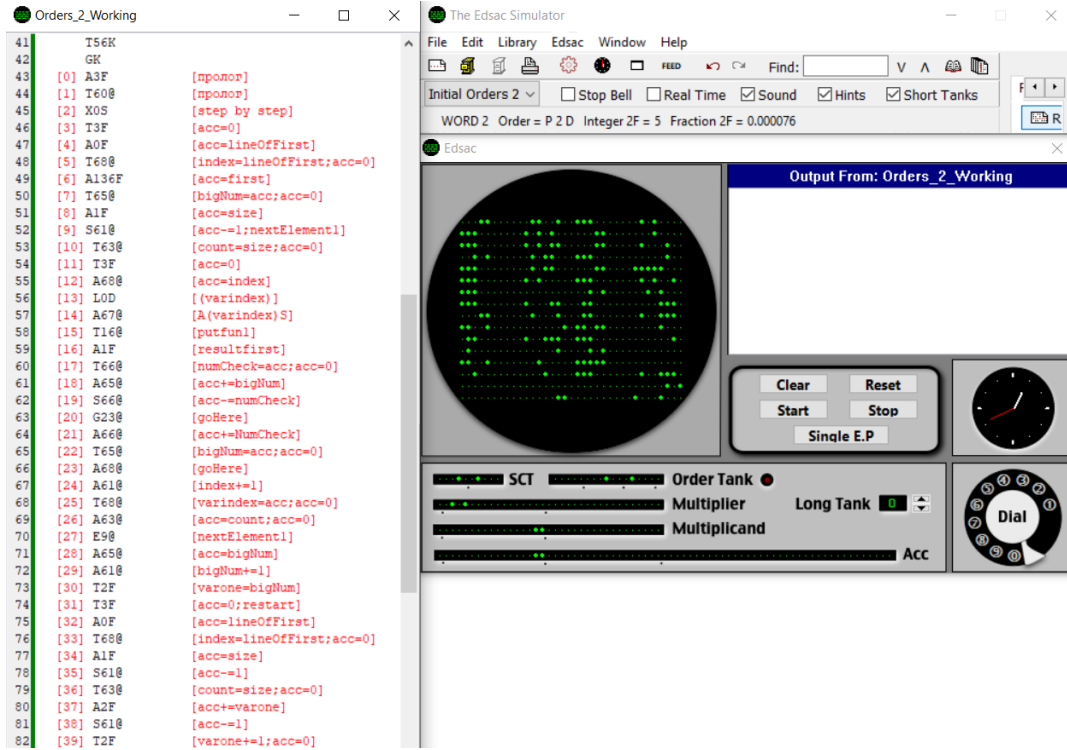
Это переменные алгоритма.

4. Работа программы Orders 1



Ответ сохраняется в строке 91.

5. Программа Orders 2



[Реализовать нахождение наибольшего общего делителя (НОД) для массива чисел при помощи EDSAC; ответ сохраняется в ячейка 2]

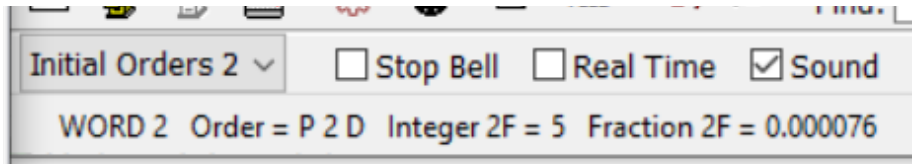
T56K

GK

[0] A3F	[пролог]
[1] T60@	[пролог]
[2] X0S	[step by step]
[3] T3F	[acc=0]
[4] A0F	[acc=lineOfFirst]
[5] T68@	[index=lineOfFirst;acc=0]
[6] A136F	[acc=first]
[7] T65@	[bigNum=acc;acc=0]
[8] A1F	[acc=size]
[9] S61@	[acc-=1;nextElement1]
[10] T63@	[count=size;acc=0]
[11] T3F	[acc=0]
[12] A68@	[acc=index]
[13] L0D	[(varindex)]
[14] A67@	[A(varindex)S]
[15] T16@	[putfun1]
[16] A1F	[resultfirst]
[17] T66@	[numCheck=acc;acc=0]
[18] A65@	[acc+=bigNum]
[19] S66@	[acc-=numCheck]
[20] G23@	[goHere]
[21] A66@	[acc+=NumCheck]
[22] T65@	[bigNum=acc;acc=0]
[23] A68@	[goHere]
[24] A61@	[index+=1]
[25] T68@	[varindex=acc;acc=0]
[26] A63@	[acc=count;acc=0]
[27] E9@	[nextElement1]
[28] A65@	[acc=bigNum]
[29] A61@	[bigNum+=1]
[30] T2F	[varone=bigNum]
[31] T3F	[acc=0;restart]
[32] A0F	[acc=lineOfFirst]
[33] T68@	[index=lineOfFirst;acc=0]
[34] A1F	[acc=size]
[35] S61@	[acc-=1]
[36] T63@	[count=size;acc=0]
[37] A2F	[acc+=varone]
[38] S61@	[acc-=1]
[39] T2F	[varone+=1;acc=0]
[40] G60@	[nextElement]
[41] T3F	[acc=0]
[42] A68@	[acc=index]
[43] L0D	[(varindex)]

[44] A67@	[A(varindex)S]
[45] T46@	[putfun1]
[46] A1F	[resultfirst]
[47] S2F	[subtract;acc=0-10]
[48] E47@	[acc-=varone;acc=-10]
[49] A2F	[checkacc+=varone;acc=0]
[50] S61@	[acc-=1;acc=-1]
[51] E31@	[acc==0;acc=-1]
[52] T3F	[acc=0]
[53] A61@	[acc+=1]
[54] A68@	[acc+=varindex]
[55] T68@	[varindex=acc;acc=0]
[56] A63@	[acc=count]
[57] S61@	[acc+=1]
[58] U63@	[count=acc;acc=count]
[59] E40@	[nextElement]
[60] E0F	[заканчиваетработу]
[61] P0D	[valone=1]
[62] P0D	
[63] P0F	[varcount=0]
[64] P0F	[varanswer=0]
[65] P0F	[varbigNum=0]
[66] P0F	[varnumCheck=0]
[67] A0F	[fun1]
[68] P0F	[varindex]
GK	
[0]A11@	
[1]T0F	
[2]A9@	
[3]T1F	
[4] A10@	
[5] T2F	
[6] A6@	[\вызов]
[7] G56F	[/подпрограммы]
[8] Z0F	[остановка]
[9] P1D	[size ;3]
[10] P2D	[varone = 1]
[11] P68F	[listStart]
[12; array] P5F	[first] [2*3 + 0 = 6]
[13] P2D	[second] [2*3 + 1 = 7]
[14] P7D	[third] [2*3 + 0 = 8]
EZPF	

6. Работа программы Orders 2



Ответ сохраняется в ячейка 2.

7. Результат

Edsac считается первым компьютером, который может выполнять широкий спектр вычислительных задач. Возможности алгоритма очень просты, но, несмотря на это, он позволяет пользователю программировать все, что требуется. Самая большая проблема с edsac заключается в том, что пользователь не будет знать, работает программа или нет, пока не будет написан полный алгоритм. Если бы комментарии внутри алгоритма можно было распознать, edsac был бы намного более удобным для пользователя.

