|  |
| --- |
| НИУ ВШЭ ФКН ПИ |
| Пояснительная записка к Микропроекту №1 |
| Работу выполнил ученик группы БПИ196 Вислович Владимир Александрович |

|  |
| --- |
| Вислович Владимир Александрович  1.11.2020 |

# Текст программы

Разработать программу, определяющую число единиц и нулей в битовом представлении заданного 24-х битового целого

# Описание алгоритма решения

В fasm не предусмотрено типа данных для 24-битного целого, поэтому будем использовать 32-битный тип данных, но отдельно проверять, что введенное число не превосходит 224-1.

Считываем число в переменную 32-битного типа, выполняя вышеописанную проверку. Предположим, что в числе у нас 24 нулевых и бита и, соответственно, 0 единичных бит, запишем данные предположения в соответствующие переменные.

Теперь, пока число не станет нулем, копируем его в отдельный регистр и применяем к копии операцию «логического И» с числом 0x1. Таким образом, если результат равен 1, то последний бит числа был единичным, иначе он был нулевым. Если бит был единичным, то увеличиваем переменную, содержащую количество единичных бит на 1 и уменьшаем переменную, содержащую количество нулей на 1.

После данной проверки побитово сдвинем исходное число на 1 вправо, чтобы перейти к проверке следующего бита.

Будем выполнять данную проверку, до тех пор, пока исходное число не станет нулем. После того, как оно станет нулем мы выводим информацию о количестве единичных и нулевых бит, содержащуюся в соответствующих переменных и завершаем программу.

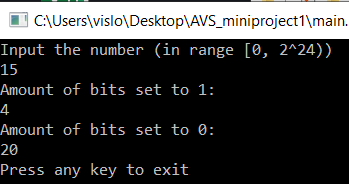
# Код программы

1. format PE console
3. include 'win32a.inc'
5. entry start
7. section '.data' data readable writable
9. ;strings to be printed later
10. numberPrompt                    db 'Input the the number (in range [0, 2^24))', 10, 0
11. strReadInt                      db '%d', 0
12. strPrintInt                     db '%d', 10, 0
13. strInvalidWarning               db 'Invalid input', 10, 0
14. printOnes                       db 'Amount of bits set to 1:', 10, 0
15. printZeros                      db 'Amount of bits set to 0:', 10, 0
16. exitPrompt                      db 'Press any key to exit', 10, 0
17. ;variables for future use
18. number                          dd 0
19. ones                            dd 0
20. zeros                           dd 24

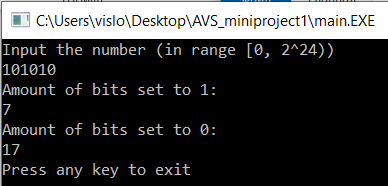
23. section '.code' code readable executable
24. ;========================= BEGIN main part BEGIN =========================
25. start:
26. ;ask to input numbery size
27. push    numberPrompt
28. call    [printf]
30. ;read number
31. push    number
32. push    strReadInt
33. call    [scanf]
34. mov     eax,    [number]
35. ;check invalid input
36. cmp     eax,    0x1000000
37. jge     invalidNum
38. check\_bit:
39. cmp     eax,     0              ;if number is 0 than end program
40. je      printFinalInfo
41. mov     edx,    eax             ;copy number to edx
42. and     edx,    0x1             ;check last bit of edx by using AND with 0x1
43. cmp     edx,    1               ;compare result of AND with 1
44. je      incOnes                 ;if it is 1 than increment amount of 1 bits (if it is 0 than do nothing)
45. shr     eax,    1               ;shift bits 1 position to the right
46. jmp     check\_bit               ;continue the loop
47. incOnes:
48. ; increase count of ones
49. mov     edx,    [ones]
50. inc     edx
51. mov     [ones], edx
52. ; decrease number of zeros
53. mov     edx,    [zeros]
54. dec     edx
55. mov     [zeros], edx
56. shr     eax,    1               ;shift bits 1 position to the right
57. jmp     check\_bit               ;continue the loop
58. invalidNum:
59. ;print that user number is too big and end the program
60. push    strInvalidWarning
61. call    [printf]
62. jmp     endProgram
63. printFinalInfo:
64. push    printOnes
65. call    [printf]
66. ;print amount of bits set to 1
67. push    [ones]
68. push    strPrintInt
69. call    [printf]
70. push    printZeros
71. call    [printf]
72. ;print amount of bits set to 0
73. push    [zeros]
74. push    strPrintInt
75. call    [printf]
76. endProgram:
77. ;ask user to press any key to exit
78. push    exitPrompt
79. call    [printf]
80. ;wait for user to press something before close console windows
81. call    [getch]
82. push    0
83. call    ExitProcess
84. ;========================= END main part END =========================
86. section '.idata' data readable import
88. library kernel, 'kernel32.dll',\
89. msvcrt, 'msvcrt.dll'
91. import kernel,\
92. ExitProcess, 'ExitProcess'
94. import msvcrt,\
95. printf, 'printf',\
96. scanf, 'scanf',\
97. getch, '\_getch'

# Тестирование программы

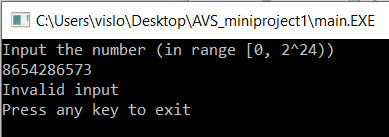
1510 = 11112



10101010 = 110001010100100102



8654286573



0

