**Student Name:** Federico Watkins

**CSIS 1800:** Introduction to Computer Science and Information Systems

**Chapter number:** 6Low Level Languages

**Assignment number:** 6

1. Describe in details at least 5 features of the Pep/8CPU that we covered in this chapter

Pep/8 Memory:

The pep/8 machine contains 65,536 bytes memory to store the data. It requires 16 bits to address each location. The specific location of the memory can be represented by using its address. The address of the pep/8 memory starts with “0000 0000 0000 0000” and ends with “1111 1111 1111 1111”. In hexadecimal representation, memory starts with “0” and ends with “FFFF”. In decimal representation, memory starts with “0” and ends with “65,535”.

Accumulator Register:

The Accumulator register is a special register in a CPU’s control unit. It can be used to perform arithmetic and logical operations. It accumulates the values in the register for each operation. At each operation, it replaces the value of accumulator by the resultant value. An accumulator holds 16 bits memory in Pep/8’s CPU. Diagrammatic representation of the Accumulator register is shown below:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A Register (Accumulator) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Instruction Register:

The instruction register (IR) is a special register in the part of CPU’s control unit. It holds the copy of the instruction currently being executed. It has 3 bytes (24 bits) of memory to process the instructions. The first 8 bits are used as instruction specifiers. The instruction specifier is used to store the binary value of the instruction to execute. The remaining 16 bits are used as operand specifiers. The operand specifier is used to store the binary value of the operand to execute. Diagrammatic representation of the instruction register:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction Specifier |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Operand Specifier |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

In the instruction specifier, the last three bits are used to specify the addressing mode of the instruction. There are two types of addressing modes:

1. Describe in details at least 5 features of the Pep/8CPU that we covered in this chapter.
2. Immediate addressing mode (000).

The operand specifier of the instruction contains the operand. In the instruction specifier, the last three bits are used to refer the addressing mode. The last three bits of the instruction specifier is “000”, then the data will be available in the operand specifier. Diagrammatic representation of immediate addressing mode: The shaded places refer to the operand and is shown below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction Specifier |  |  |  |  |  | 0 | 0 | 0 | 8-Bits |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Operand Specifier |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16-bits |

1. Direct addressing mode (001):

The operand specifier of the instruction contains the address of the data, where the data have been stored. In the instruction specifier, the last three bits are used to refer the addressing mode. If the last three bits of the instruction specifier is “001”, then the address of the data will be available in the operand specifier. And the data will be available in the specified address. Diagrammatic representation of direct addressing mode: The shaded places refer to the operand and is shown below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction Specifier |  |  |  |  |  | 0 | 0 | 0 | 8-Bits |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Operand Specifier |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16-bits |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16-bits |

Program Counter:

The Program Counter (PC) is a special register in a CPU’s control unit. It contains the address of the next instruction that has to be executed.  After executing the instruction in the instruction register, the control will move to execute the instruction from the address in the program counter. It has 2 bytes (16 bits) of memory to process the instructions. Diagrammatic representation of the Program Counter is shown below:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Program Counter |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16-bits |

1. By assuming that a minimal allocation is a word = 2 bytes, calculate how many bits are required to address the memory having total capacity of:
2. 64 Kbytes

Word = 2-bytes = 16-bits

64-Kbytes = 524288-bits

524288 ÷ 16 = 32768

Hence, to address the memory having total capacity of said quantity it would require 32768-bits.

1. 16 Mbytes

Word = 2-bytes = 16-bits

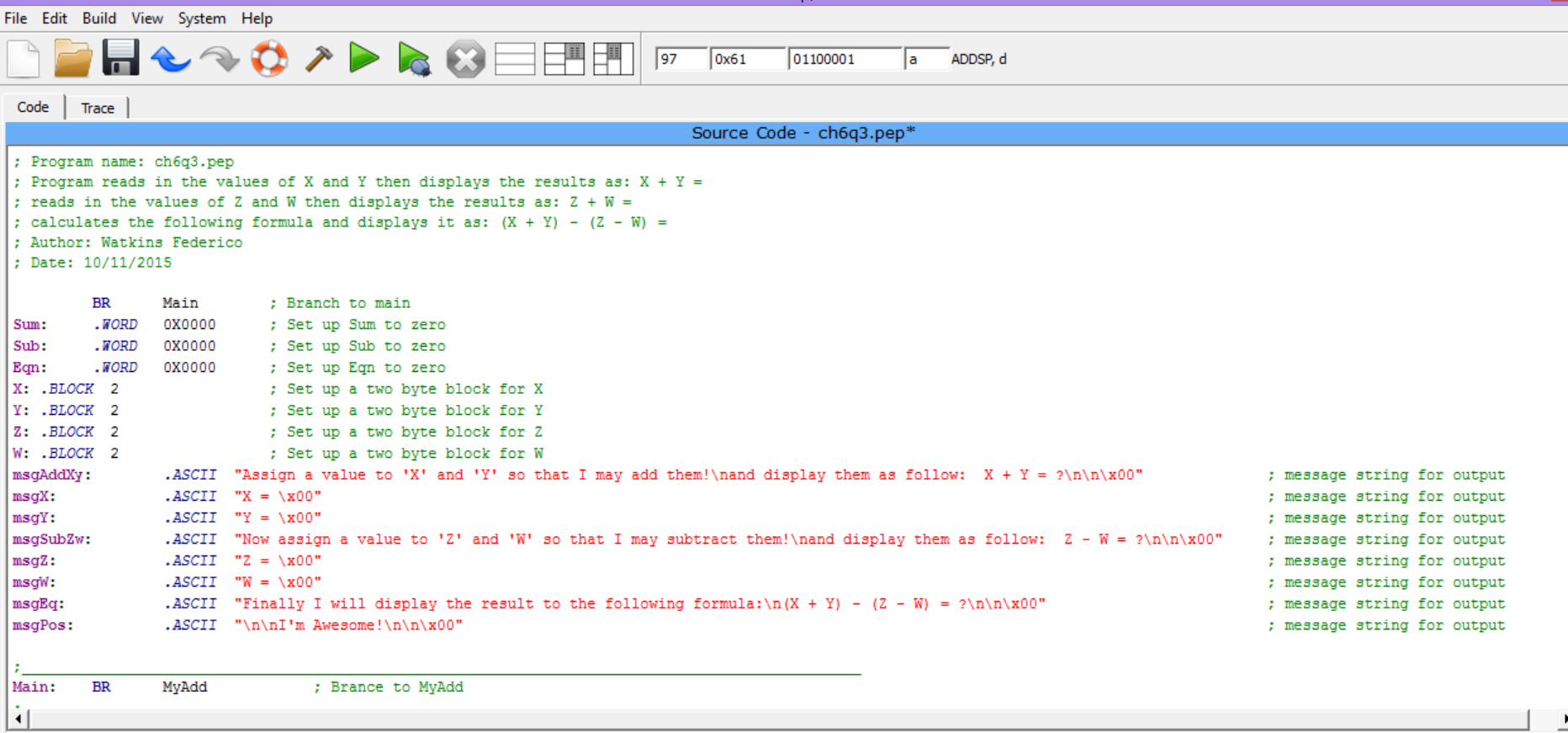
16-Mbytes = 134217728-bits

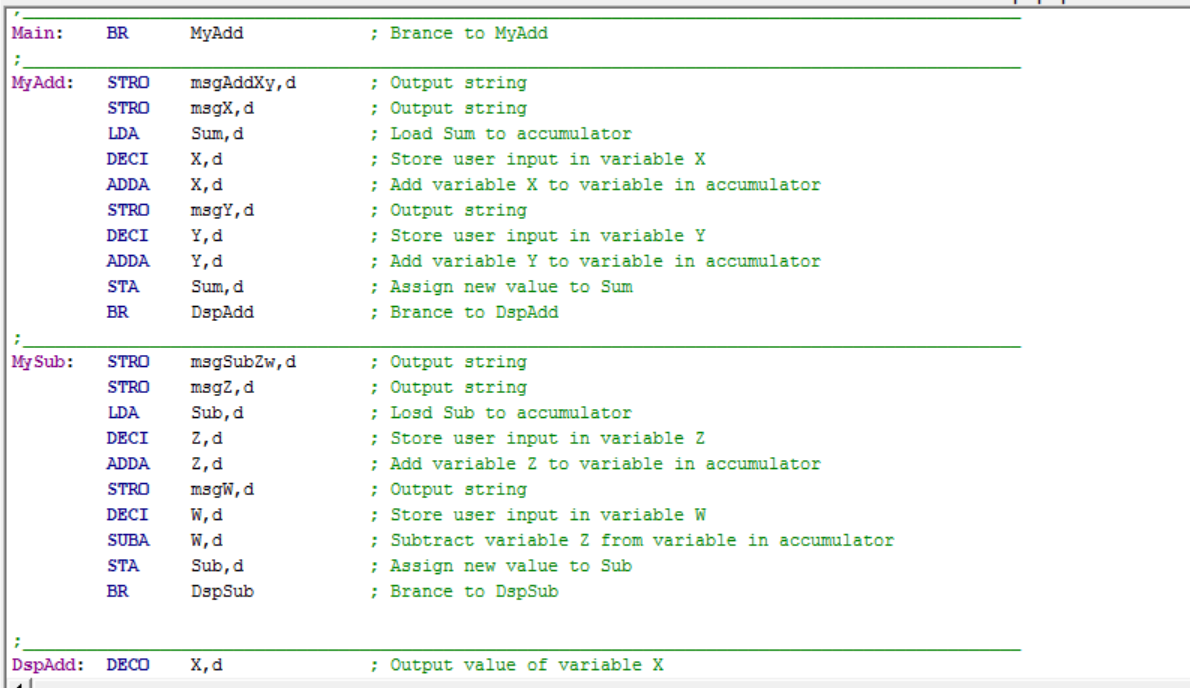
134217728 ÷ 16 = 8388608

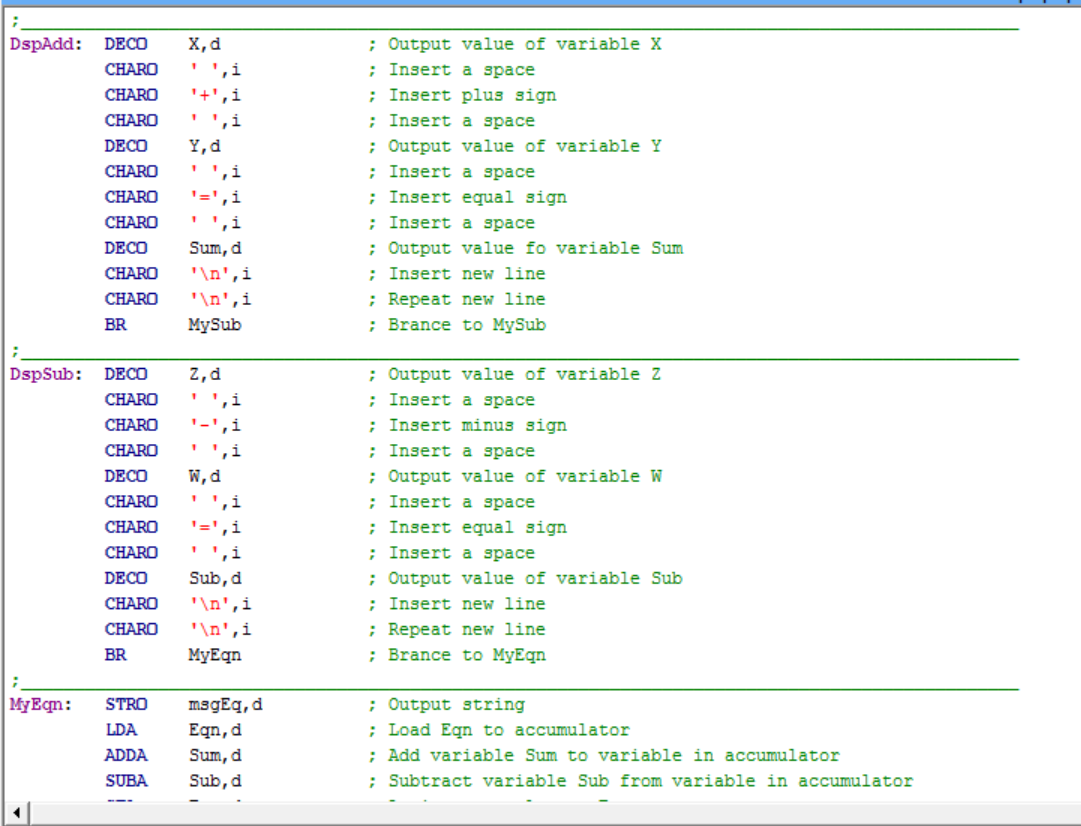
Hence, to address the memory having total capacity of said quantity it would require 8388608-bits.

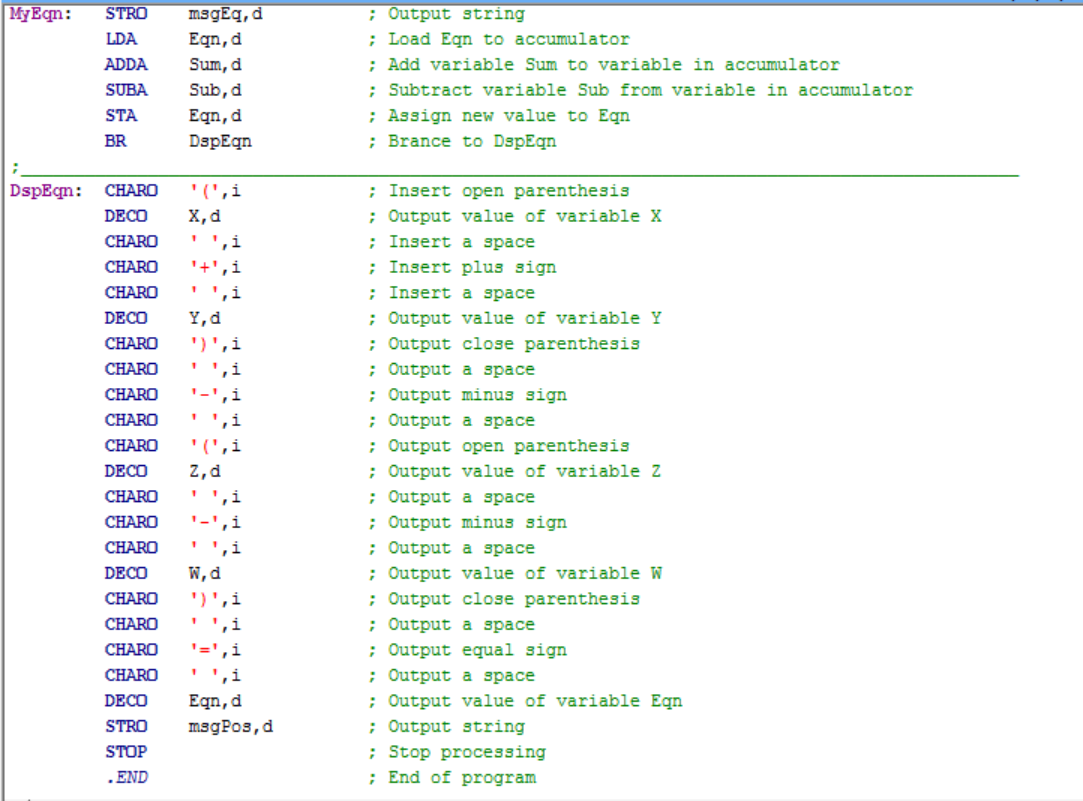
1. Write a program in Pep/8assembly that:

* Reads in the values of X, and Y the then displays the results as: X + Y =
* Reads in the values of Z, and Z the then displays the results as: Z - W =
* Calculated the following formula and displays it as: (X + Y) - (Z - W) =

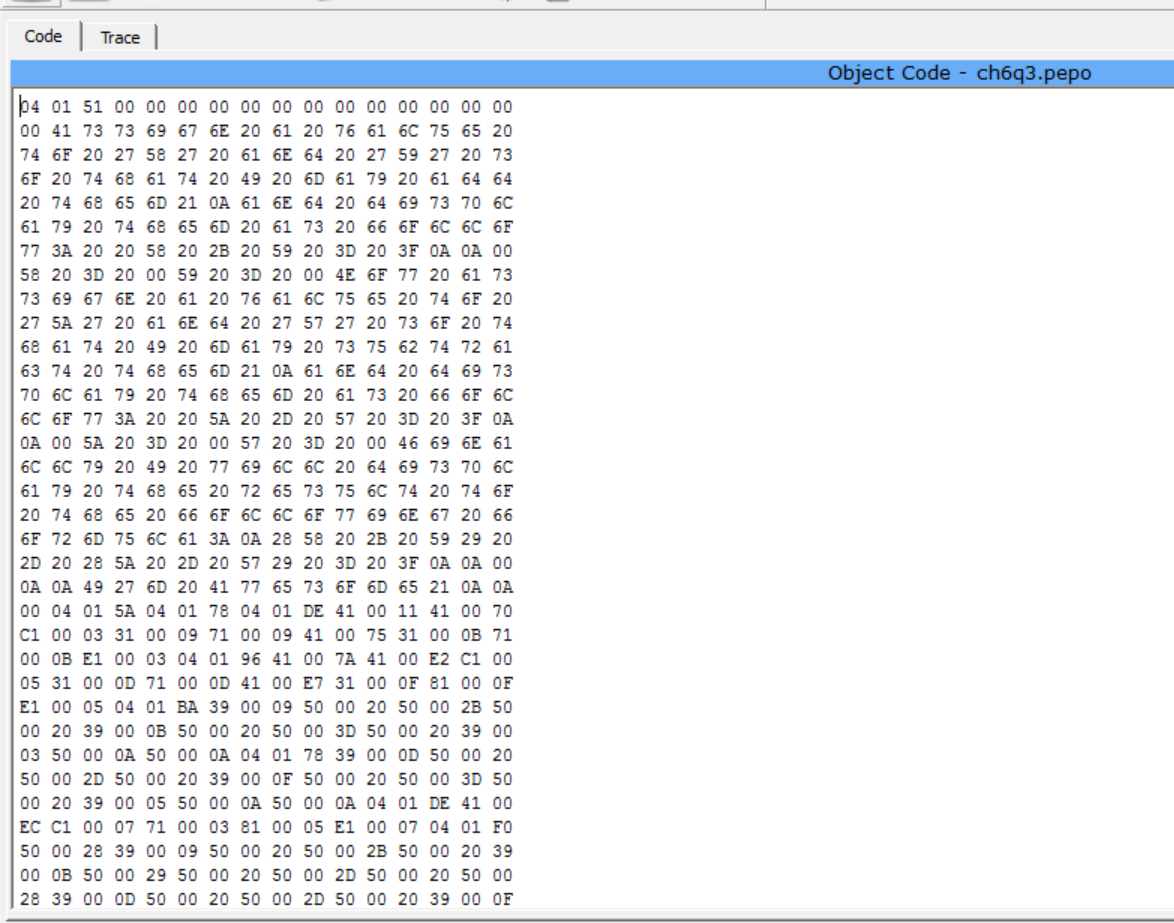


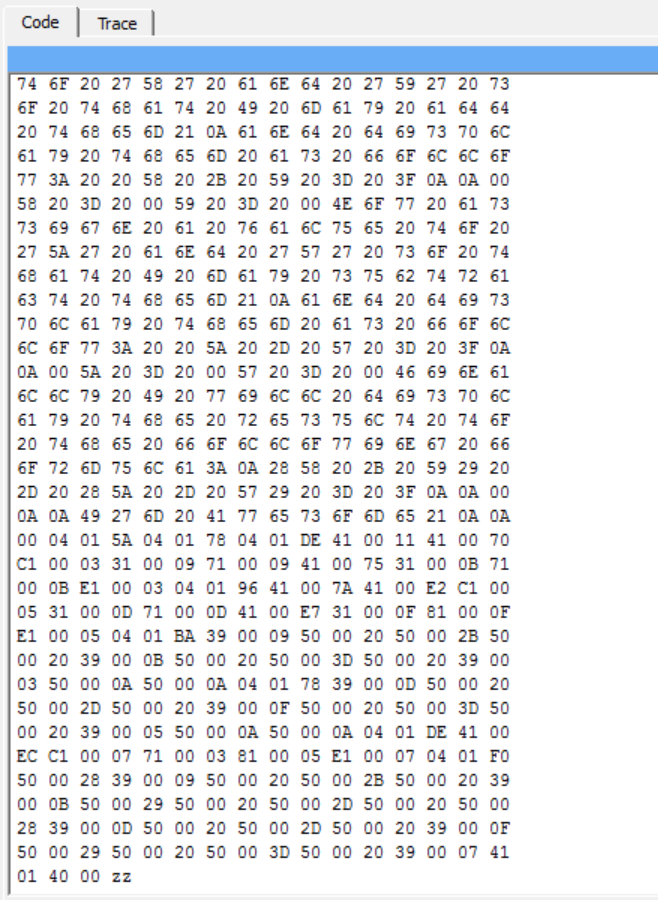




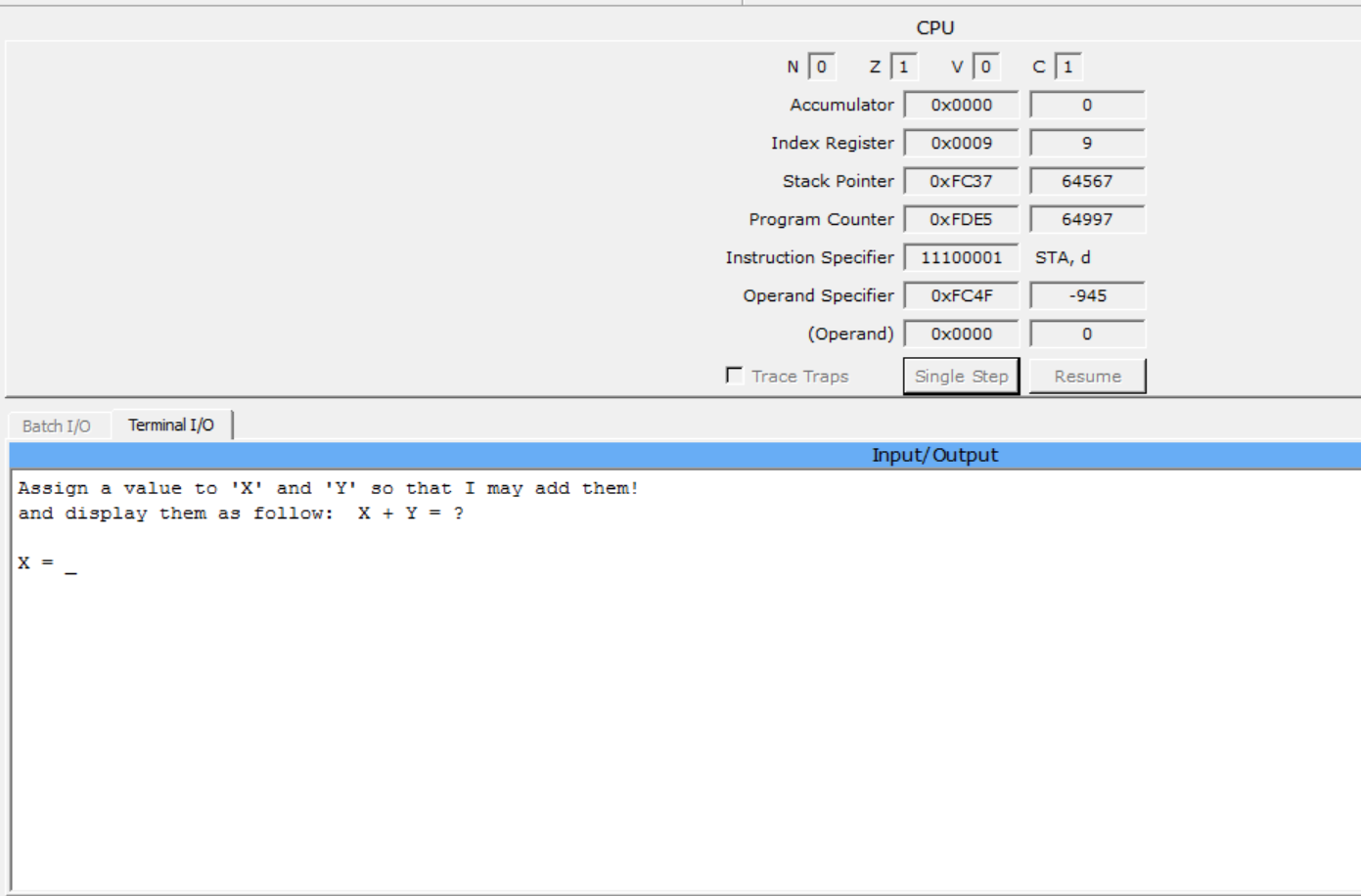


Source Code (comments in code)

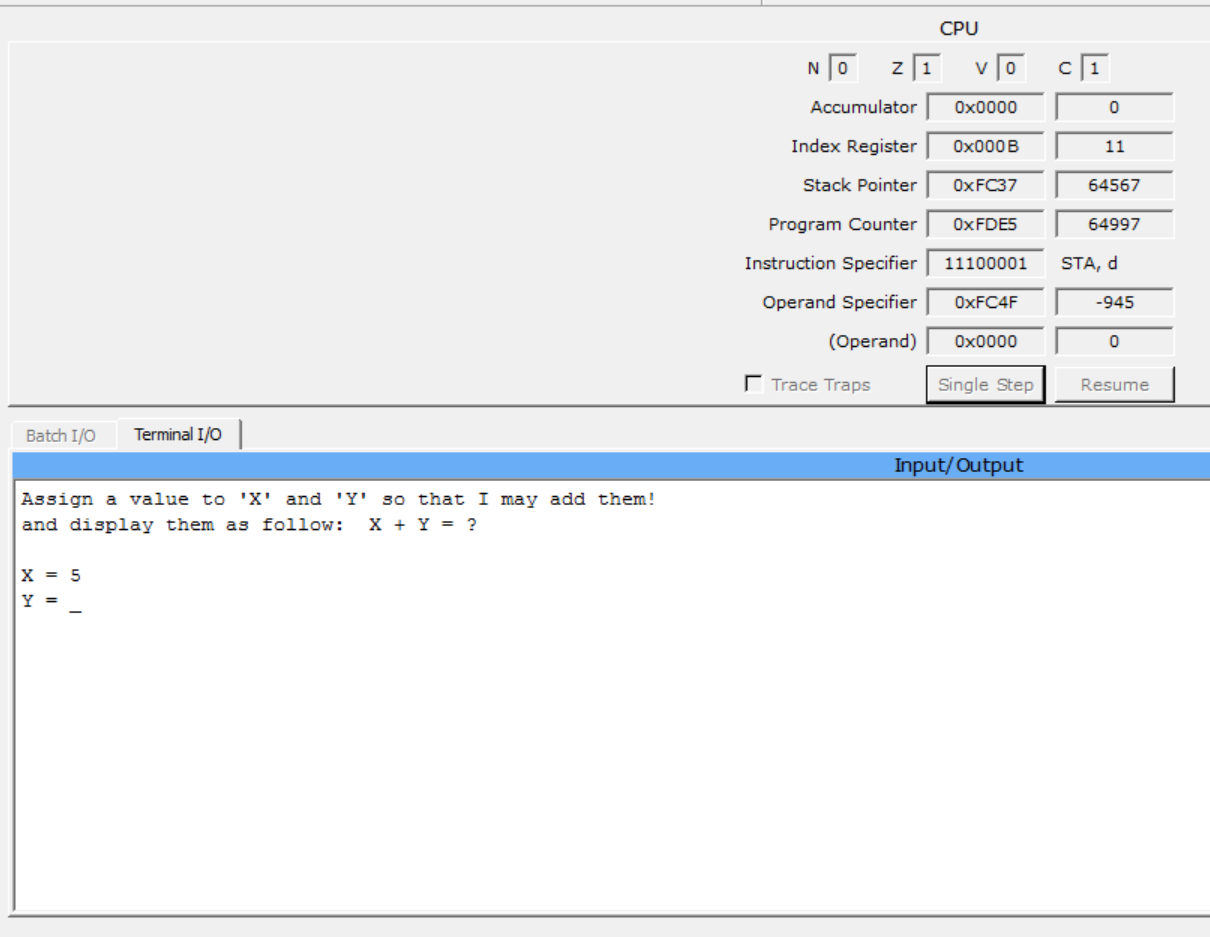




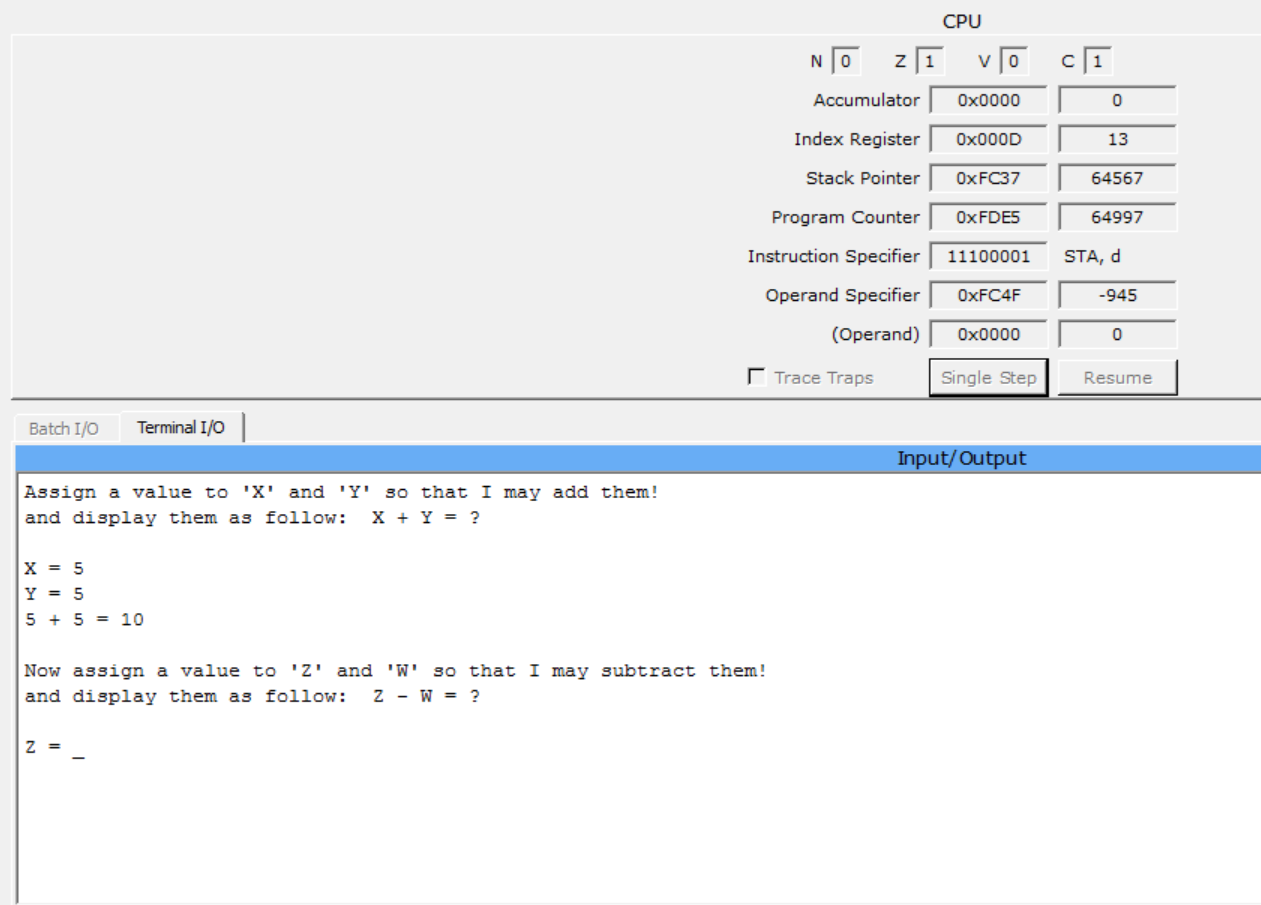
Object Code



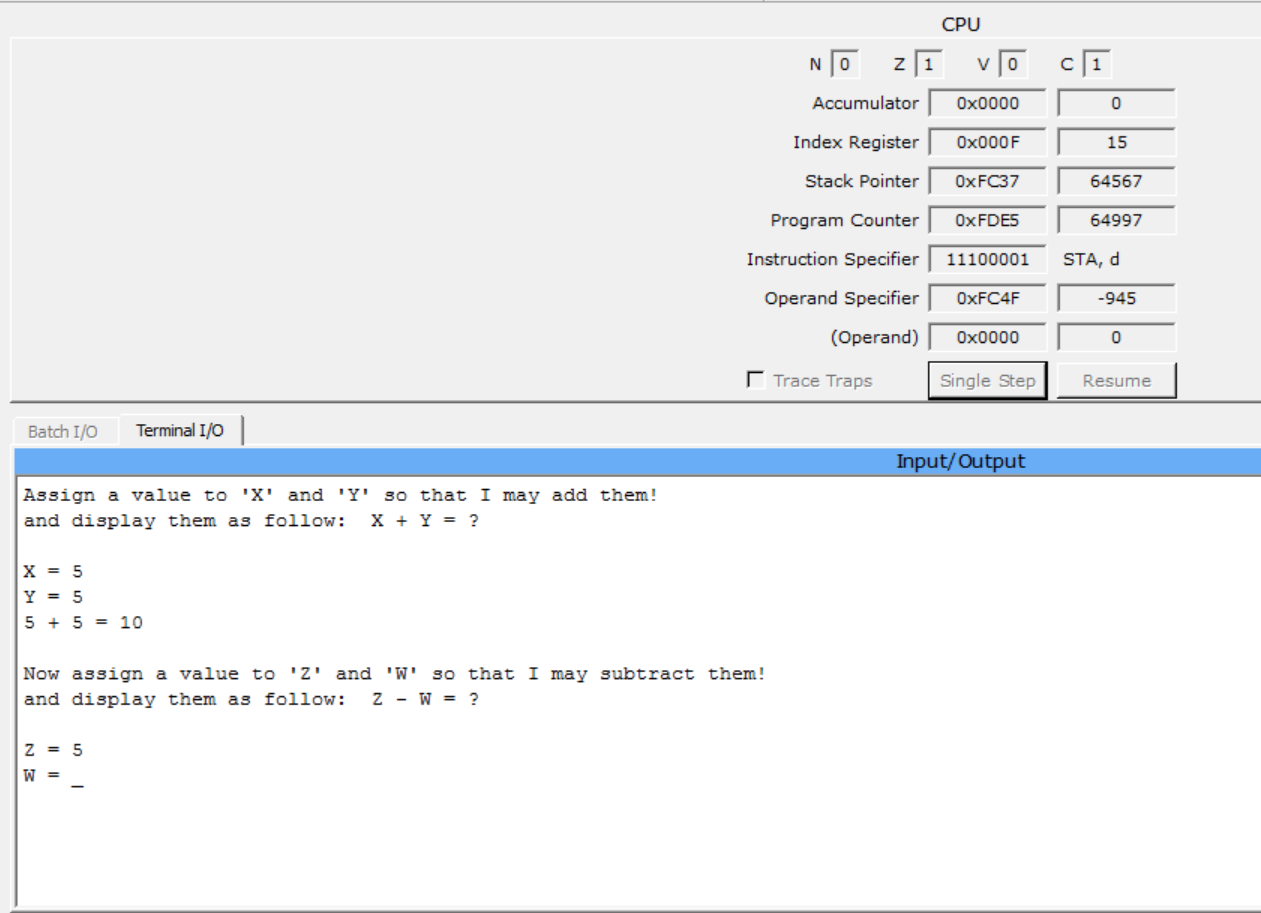
Program Starts



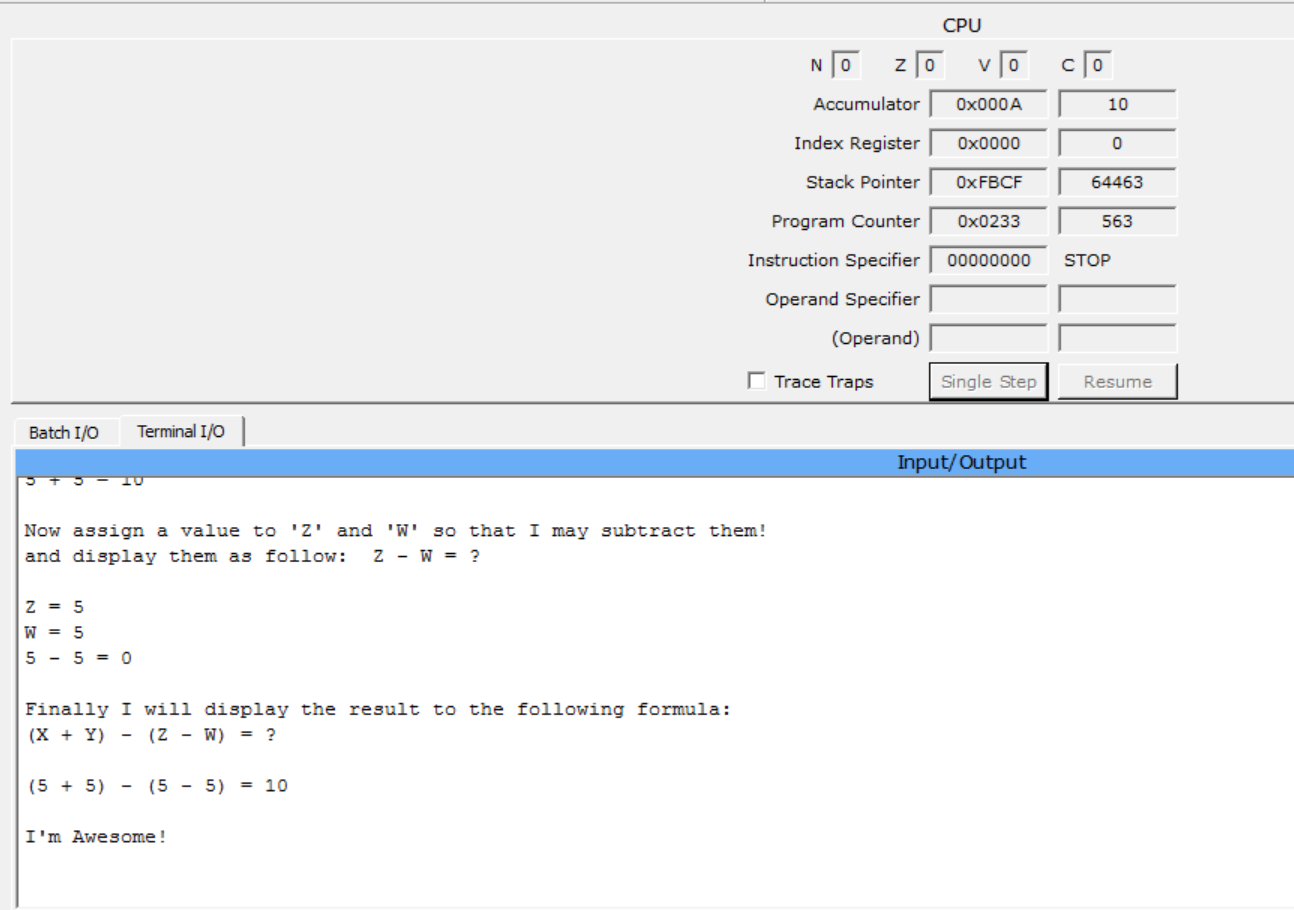
Enter ‘5’ as input for X



Enter ‘5’ as input for Y. Operation and result are presented in requested format.



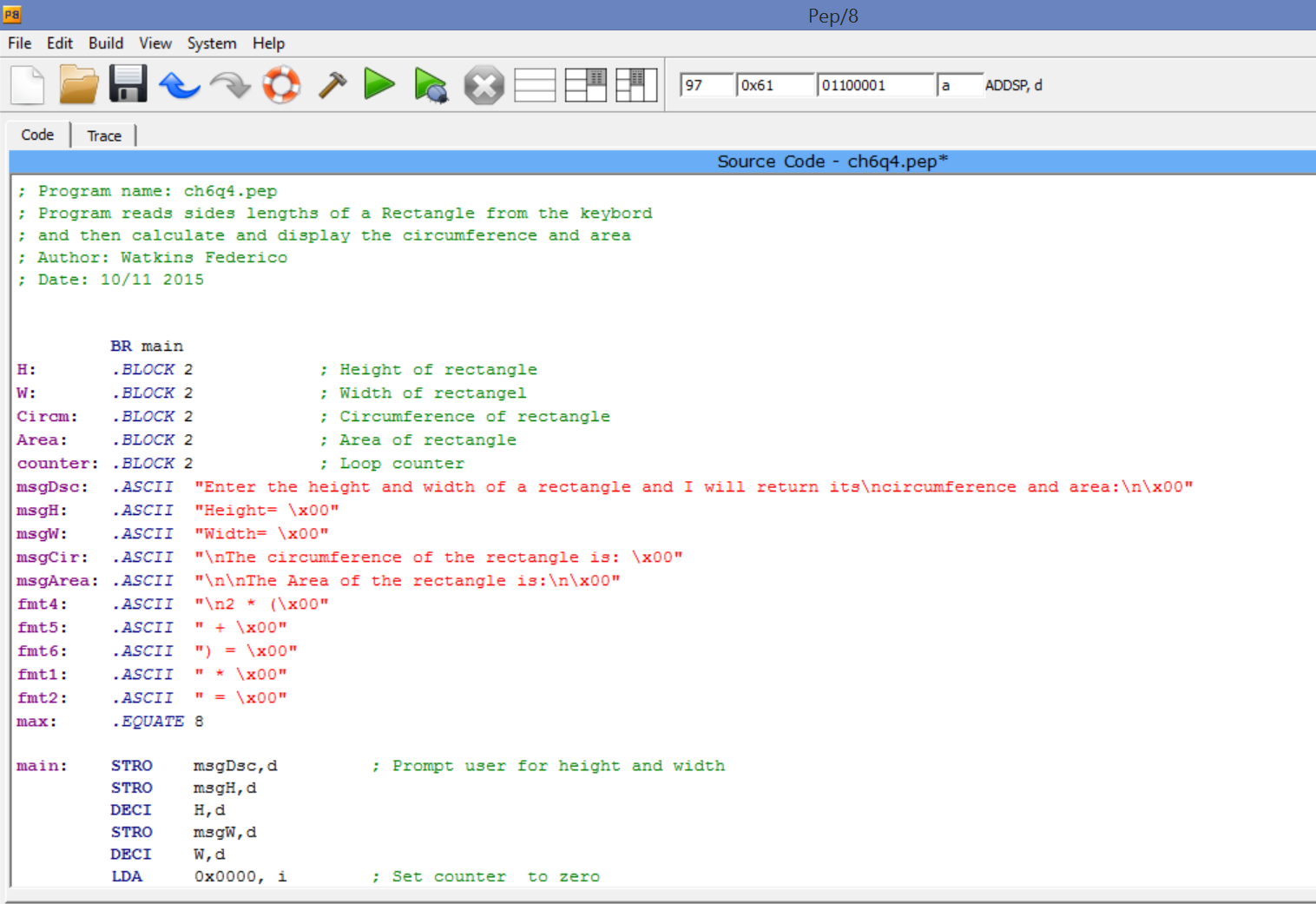
Enter ‘5’ as input for Z

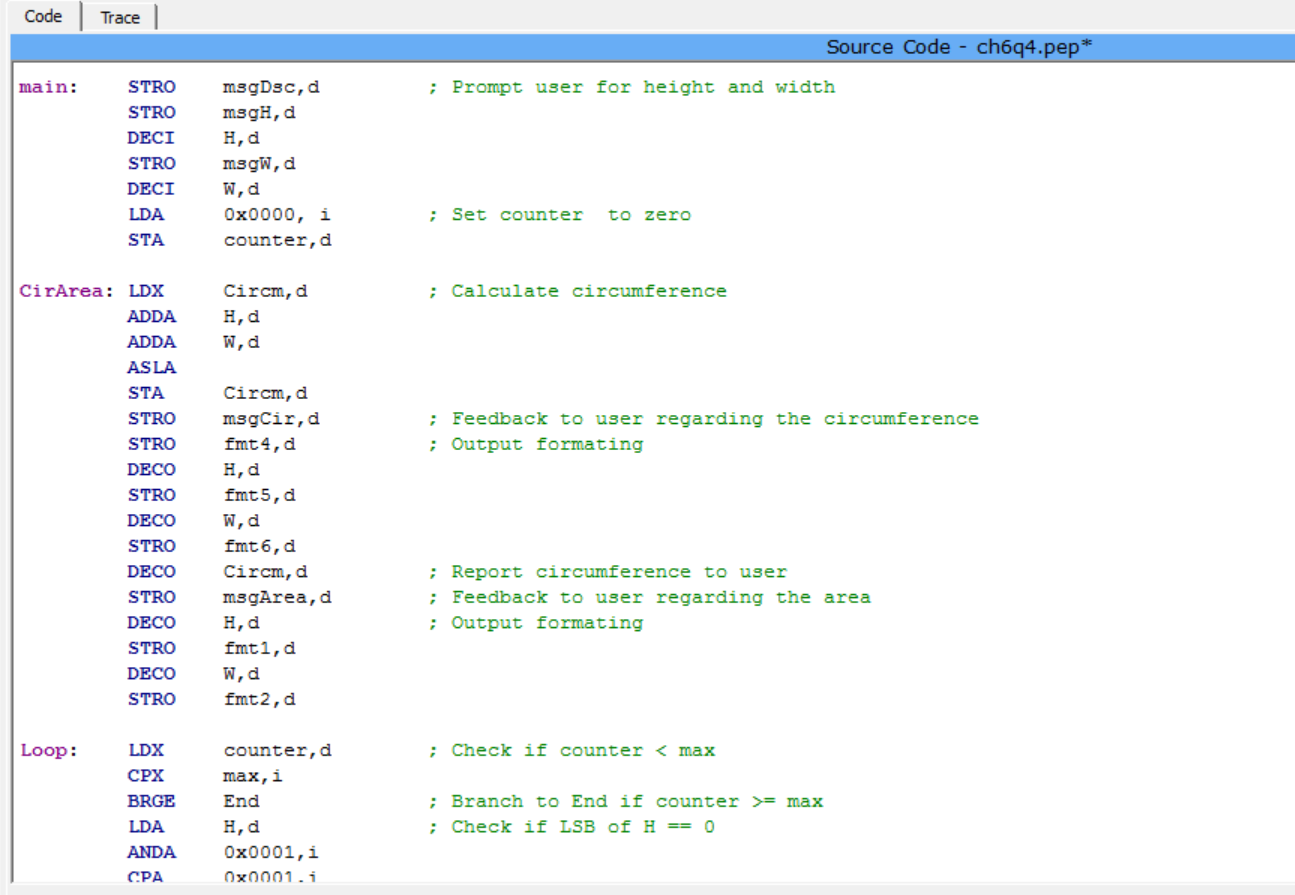


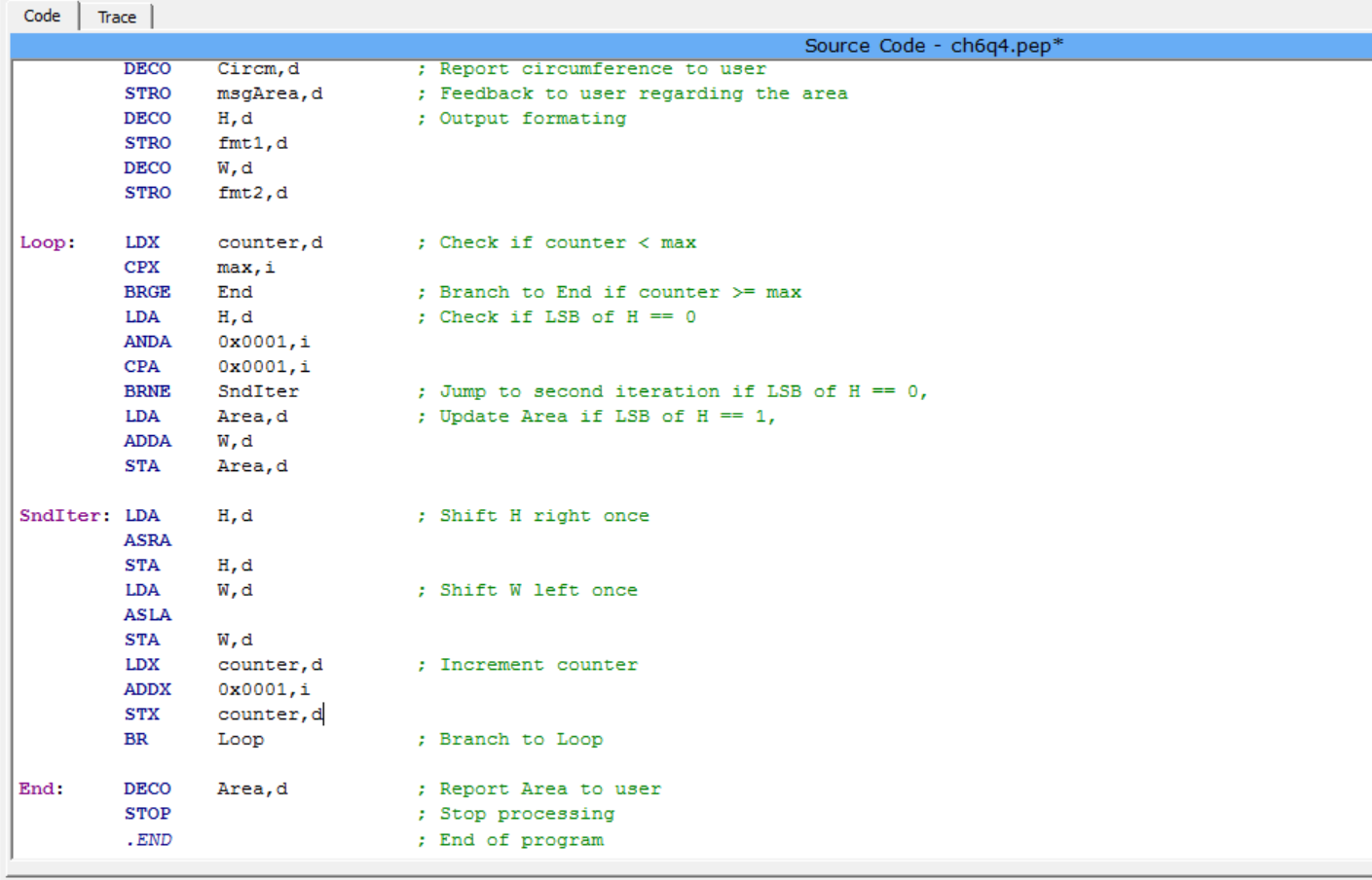
Entered ‘5’ as input for W. Operation, result and equation are presented in requested format.

Run

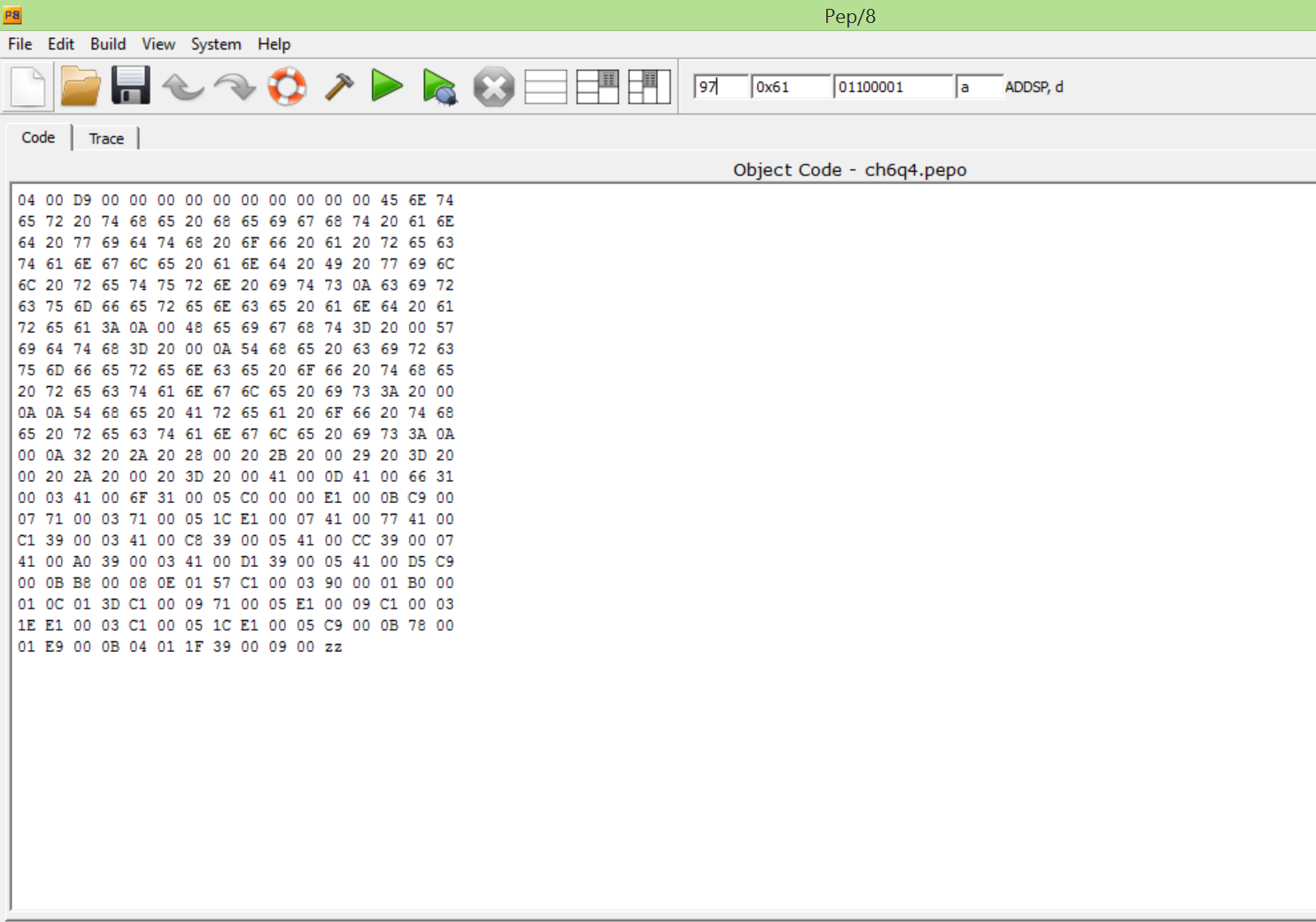
1. Write a program to read sides lengths of a Rectangle from the keyboard and then calculate and display the circumference and area of it. The display must be informative!



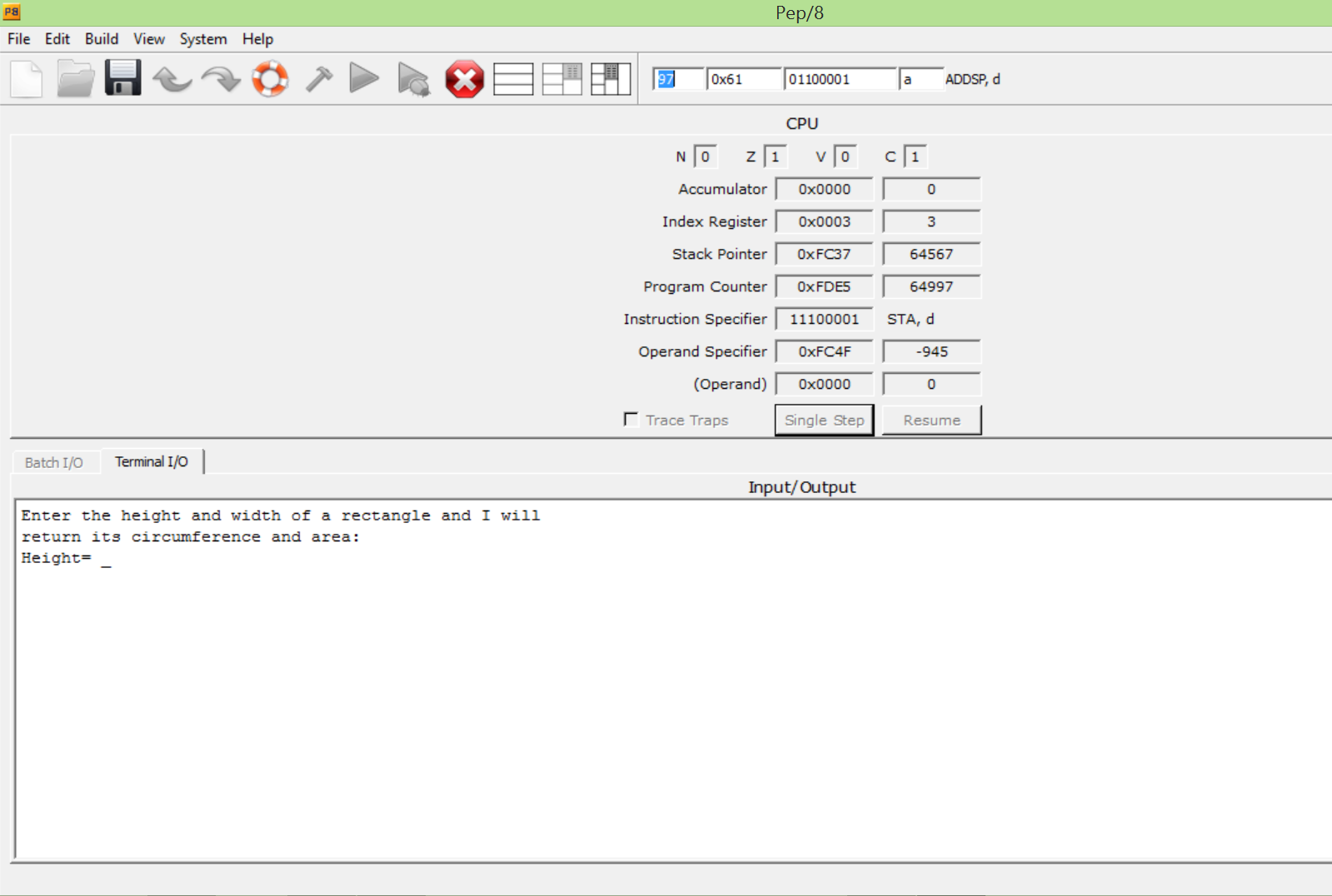




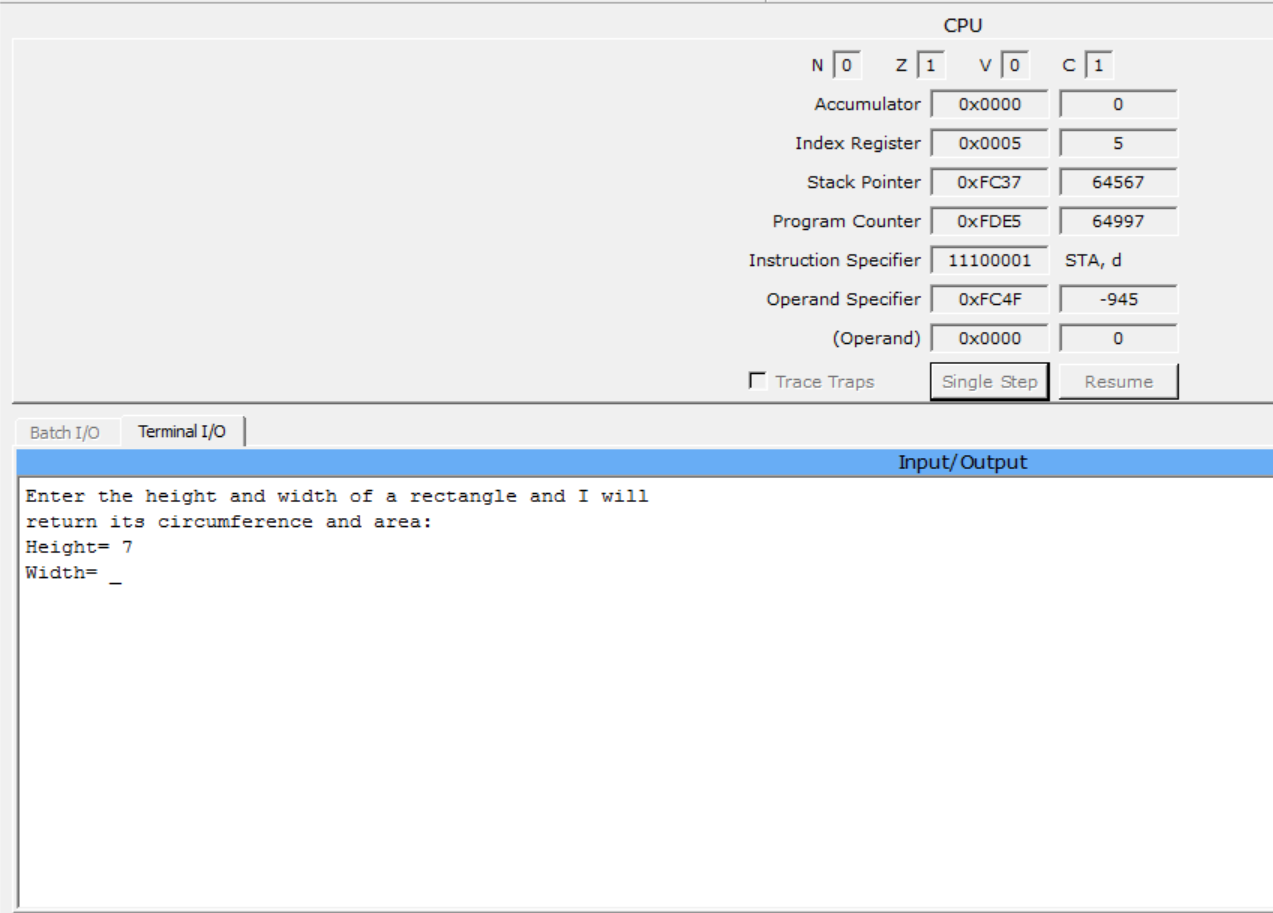
Source code



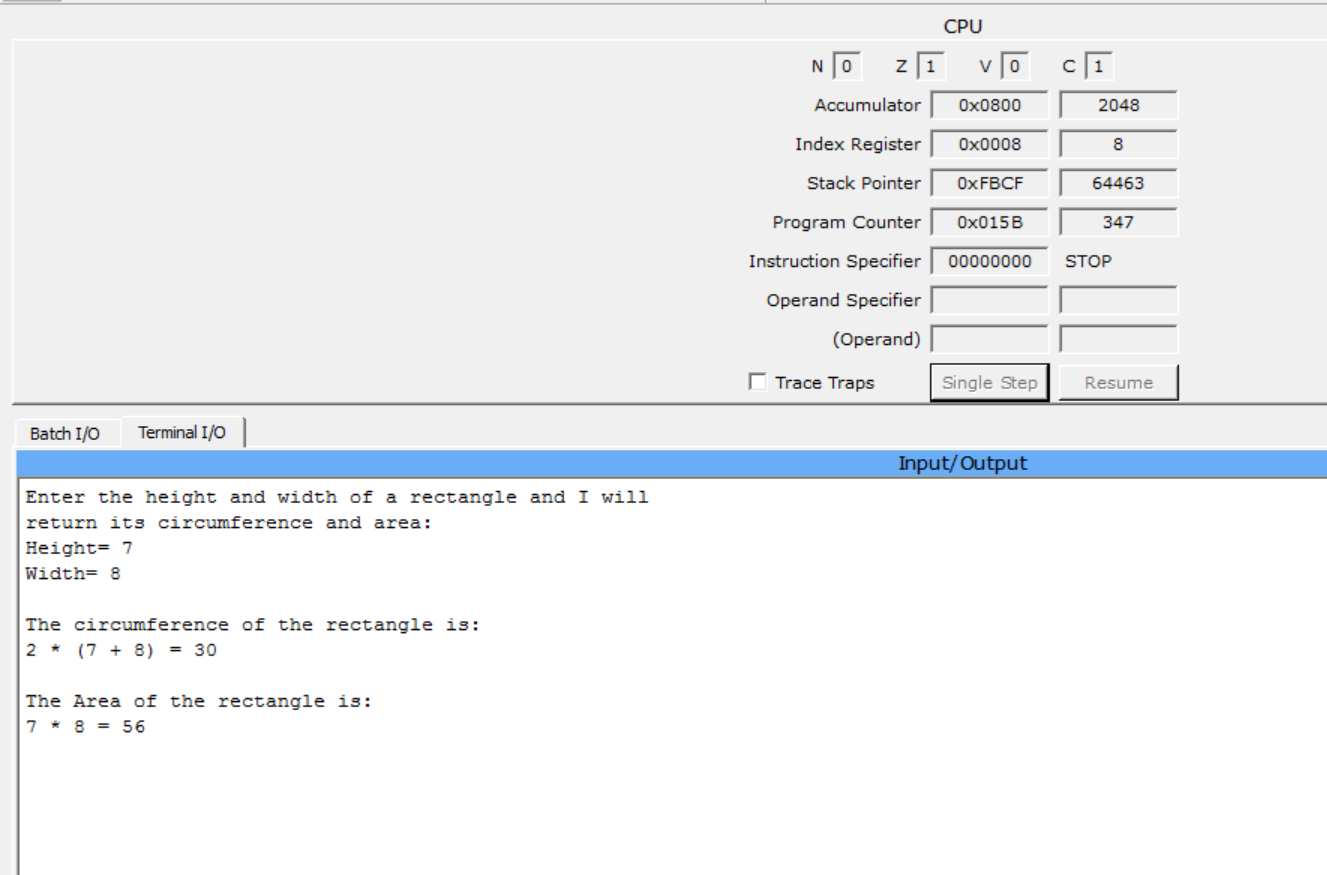
Object code



Program starts



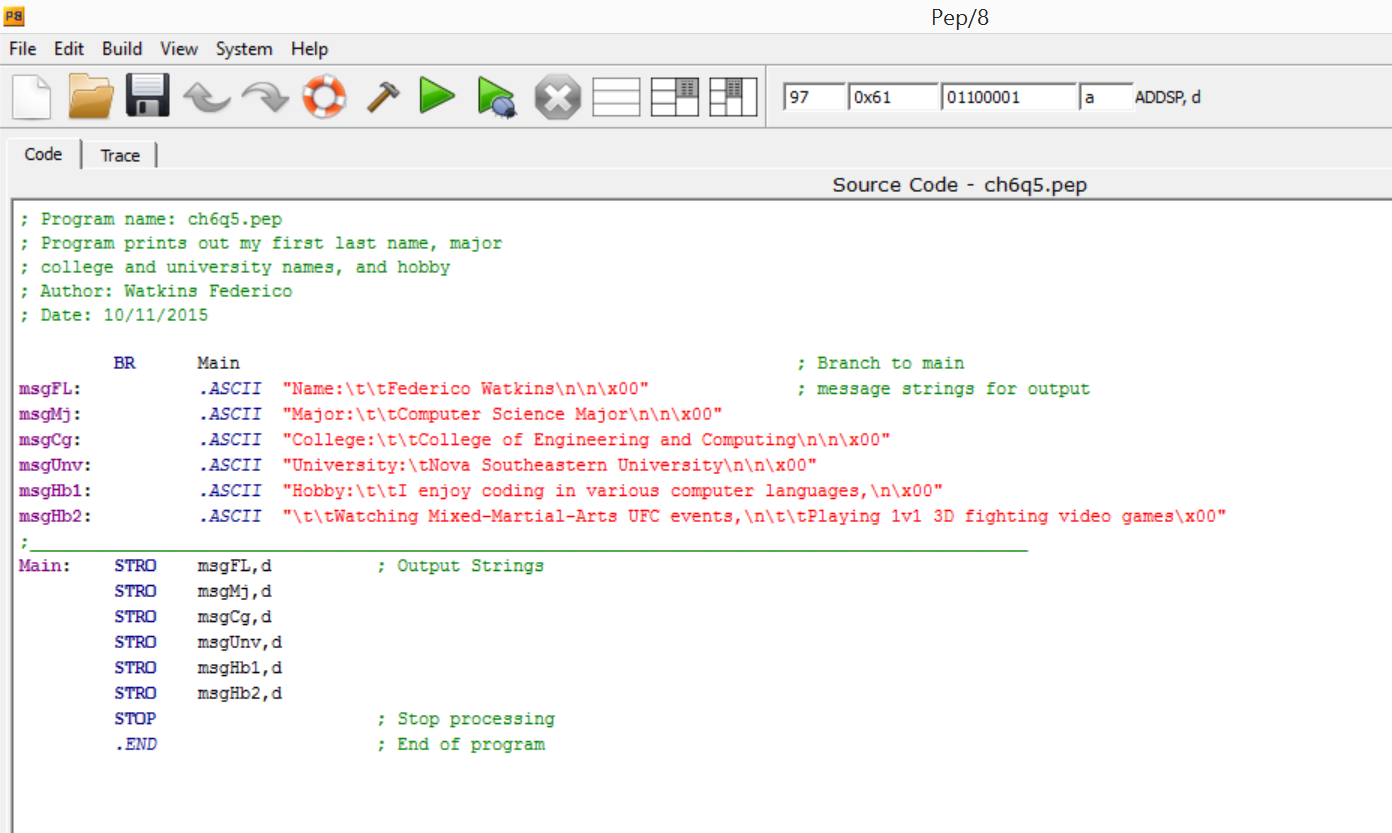
Entered ‘7’ as input for Height.



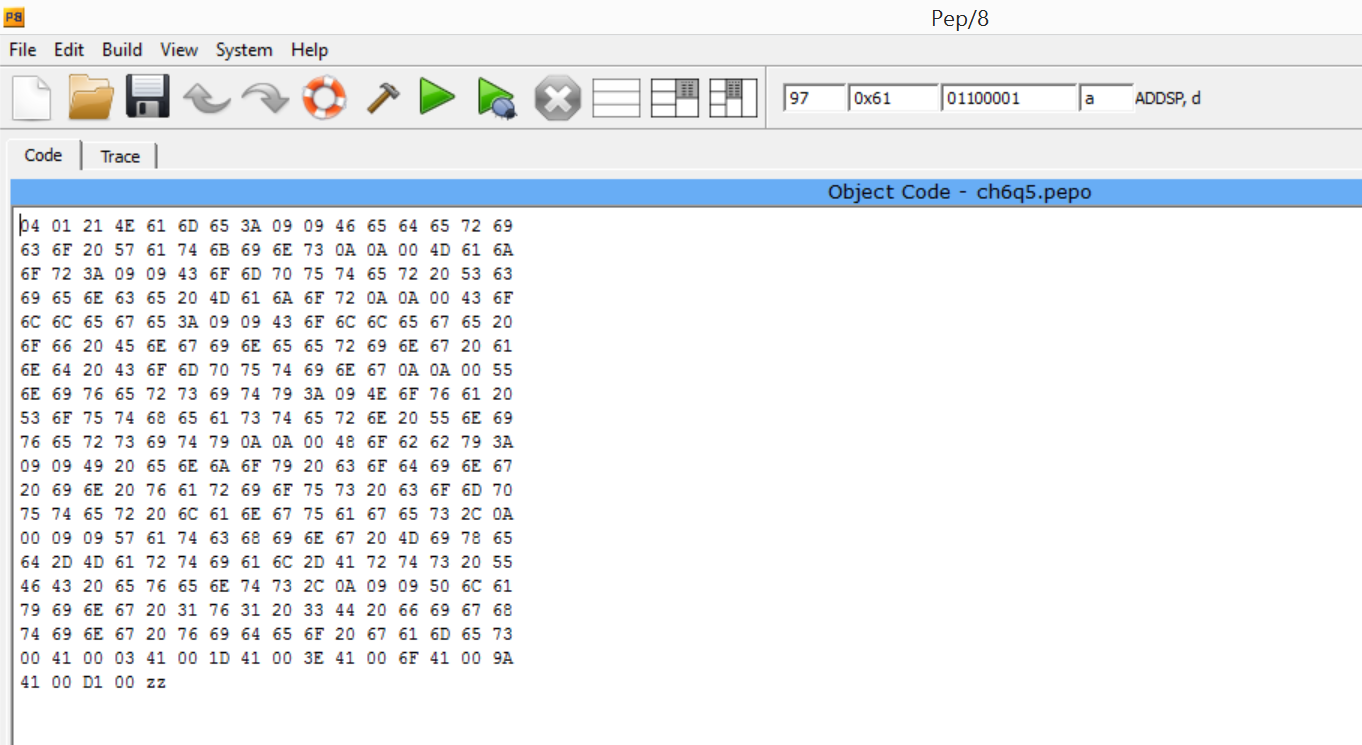
Entered ‘8’ as input for Width. Operation and result of Circumference (Perimeter) and Area are presented in requested format.

Run

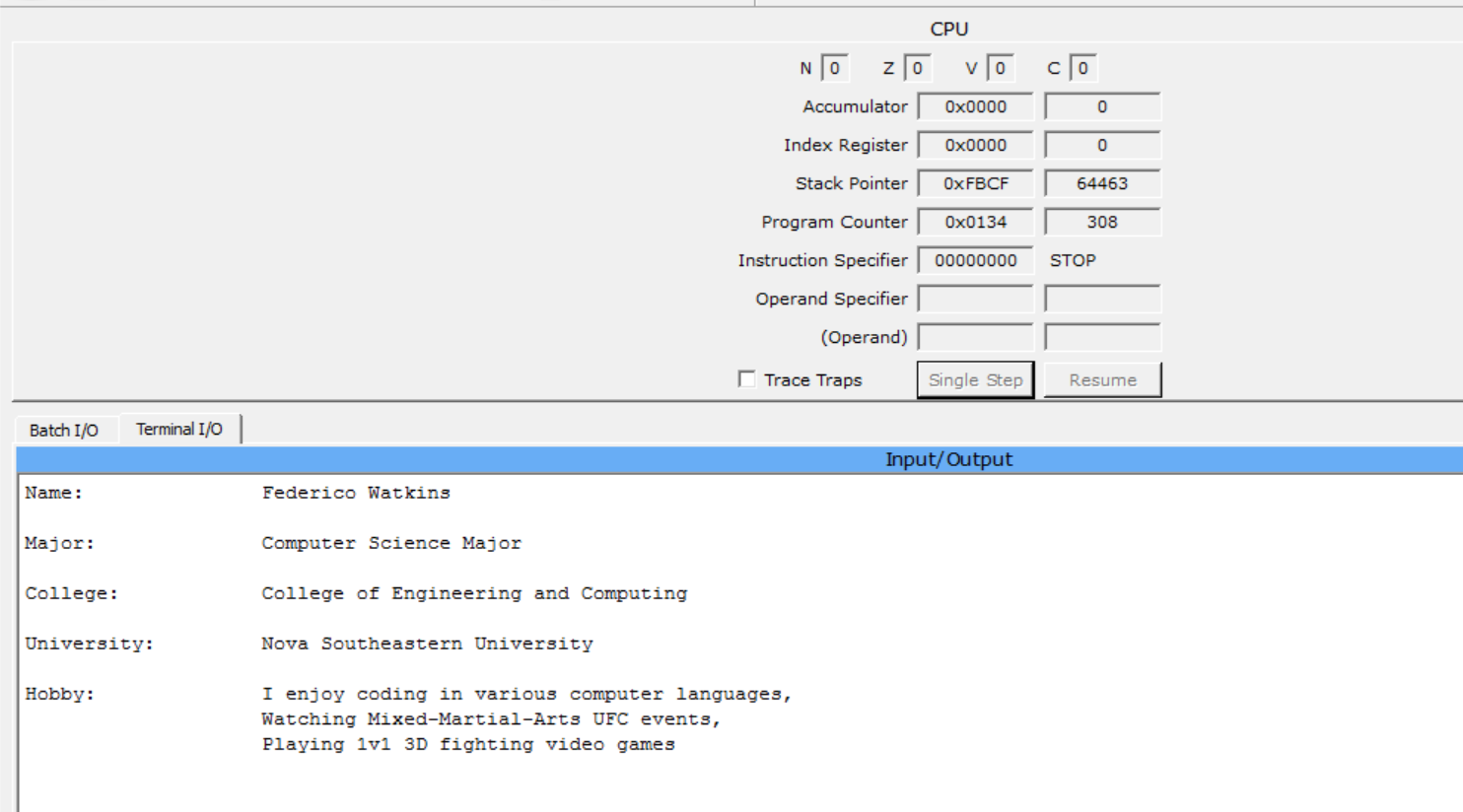
1. Design a program for writing out your FIRST LAST name, your major, college and university names and your hobby given that the implementation language is Pep/8 assembly language. Use string (not character) manipulation instructions.



Source code



Object code

  
Program displays requested information in the requested format.

Run

References:

1. Dale and Lewis, *Computer Science Illuminated*, p. 151-187
2. https://code.google.com/p/pep8-1/