Halil BAYDAR - 2017400297

I used c++ programming lenguage thanks to pointer facilities and my project consists of only one main.cpp file. I have seven variables in the type of unsigned short which represents seven 16 bit registers and fifteen pointers that point their addresses with according to their types: For example, unsigned char pointer points the most 8 significant bit of 16 bit register and unsigned short pointer points the all bit of 16 bit register. I have multiple arrays to check the given input to figure out what it is register or register with breaked or not. There is a variable array that keep the name and type of given variables and named as index keeper array to holds index of variables in the memory. I have tokenizer helper function to reads the given input properly. It returns line by line given code to process to the line. In first of all, I reads all given input to put whole code and also variables into the memory. And I also read whole input again to process all given code with regard to the first string in per line. Mov, Add and Sub functions work similar, in the first step they traverse variable array to find out if variable is in the process or not and then they check the destination operand that in the right hand side of instruction and in the left hand side of comma is direct register addresing or indirect register addresing operand and then check memory addresing operand and finally controls the destination operand whether it is variable or not to continue the process. According to the destination operaind some boolean variables are changed and there is a helper funciton which returns the value of the right hand side of the comma by eliminating stuffs and calculating requested value and which works with regard to them. Mul and Div functions also works similar because there is only source operand in the line and this functions checks it is register direct or register indirect or memory direct operand and also variable or immediate operand to get into the process. At the end of the process I modify flags according to the result of operatin and if there is overflow in the div operation this code gives an error message with the code line in the input. And, xor and or instructions also work similar, so I checked what the operand is register direct addresing, register indirect addresing, memory addresing or variable in the memory ,and then thanks to return_value_from_right_hand_side helper functon, this code puts left operand into the process with the value and flags ar changed. Rcr,rcl,shl,shr functions are same in terms of left operand and they can be processed with cl register and immediate value lower than 31d number so return_value which is helper function returns the value and it is processed with it. Inc and dec functions controls destination operands like other I mentioned and increases or decreases the destination operand by one and at the end changes flags according to flags. Push function works only with word size operand and put the value into the memory by splitting the most 8 significant and the least 8 significant bits of the value. Cmp function controls destination operand like other funcitons and changes flags. Jmp functions traverse code array that keeps whole given code to find label and checks the related flags and it equlizes iterator with label index and it jumps. Int 21h function only checks ah register if it has 01, the funciton reads the input then puts it into al register ,and if it has 02 this function writes dl ascii value to screen. If iterator reaches to the int 20h function, this program ends up.