Explanation of the program:
This program takes some hexadecimal inputs from the user, then makes some operations such as addition, division etc., and prints out the result.

adding:
When program jumped to this label, program pops first two elements in the stack as ax and cx, then add them together. After adding operation, it pushes the sum as ax. Finally program jumps to take_input label.

multiple:
When program jumped to this label, program pops first two elements in the stack as ax and cx, then multiply these two values. After multiplying operation, it pushes the result as ax. Finally program jumps to take_input label.

division:
When program jumped to this label, program pops first two elements in the stack as ax and cx. After it makes integer division operation as dividing ax by cx. Then it pushes the result as ax. Finally program jumps to take_input label.

xorxor:
When program jumped to this label, program pops first two elements in the stack as ax and cx, then makes xor operation with these two values and pushes the result as ax. Finally program jumps to take_input label.

oror:
When program jumped to this label, program pops first two elements in the stack as ax and cx, then makes or operation with these two values and pushes the result as ax. Finally program jumps to take_input label.

andand:
When program jumped to this label, program pops first two elements in the stack as ax and cx, then makes and operation with these two values and pushes the result as ax. Finally program jumps to take_input label.

dummylabel:
dummylabel label is used to jump to take_input
dummylabel2:
dummylabel2 label is used to jump to return_stack
dummylabel3:
dummylabel3 label is used to jump to letter
dummylabel4:
dummylabel4 label is used to jump to adding
dummylabel5:
dummylabel5 label is used to jump to multiple

take_input:

This label first takes input and stores it as al register. Then jumps to suitable label according to meaning of the character. (e.g if al is '+' jumps to adding label, if al is '|' jumps to oror label etc.) If input is enter it means end of the line and jumps to return_stack label. If input is not an operator or space or enter, it means it is a numerical value. Finally it jumps to letter label if character is a letter, jumps to digit label if character is a digit.

letter:
This label is used for computing the hexadecimal numerical value when the input is a letter by subtracting ascii value of char '7' from the input. Then it multiples previous digit by 10h and adds up the product with the new digit. Now the sum which is our final value is stored as bx.

digit:
This label is used for computing the hexadecimal numerical value when the input is a digit by subtracting ascii value of char '0' from the input. Then it multiples previous digit by 10h and adds up the product with the new digit. Now the sum which is our final value is stored as bx.

return_stack:
This label is called at the end of the input line. If the input line contains only one number, program jumps to setup_string label. Otherwise, it first pops the last value from the stack and then jumps to setup_string label.

space:
Space label is called when the input is space character. If the last input before the space is a number, it pushes the number, otherwise it jumps to take_input label to continue taking input process.

convert_letter:
This label is almost the reverse of the "letter" label. It converts numerical hexadecimal value to its char letter equivalent. Then it adds it to end of the string buffer and decrements the buffer by one. If buffer ends, program jumps to print_cr label, otherwise it jumps to convert_hexa.

convert_hexa:
This label is almost the reverse of the "digit" label. It converts numerical hexadecimal value to its char digit equivalent. Then it adds it to end of the string buffer and decrements the buffer by one. If buffer ends, program jumps to print_cr label, otherwise it jumps to convert_hexa.

setup_string:
It adds "$" character to the end of the buffer.

print_cr:
It prints out the carriage return before the output.

printout:
It prints out the output value.

exit:
This label terminates the program.