

HLC Broad:

Declaration

Assignment - possibly
-> 2-arg or 3-arg
arithmetic.

If -> comparison
operator statement ->
Body

Else -> Body

While -> comparison
operator statement ->
Body

Print

List of HLC:

+

-

/

*

=

>

<

>=

<=

==

!=

unsigned

signed

if

else

while

print

YMC Encoding:

add [dest. register], [first arg register] [second arg register]

sub [dest. register / first arg], [second arg register]

div [dest. register / first arg], [second arg register]

mul [dest. register / first arg], [second arg register]

movmr [address of var to assign], [current register]

jle If first operand is less than or equal to second operand[address of next instruction]

jge If first operand is greater than or equal to second operand[address of next instruction]

jl [address of next instruction]

jg [address of next instruction]

jne [address of next instruction]

je [address of next instruction]

literal: absence of - symbol, variable: sign flag from table is 0

literal: presence of - symbol, variable: sign flag from table is 1

cmp [second arg register], [first arg register] —————> use jump associated with relational operation

When else follows if statement, the if statement should jump to start of else if condition is false, and past the body of else at the end of the if statement's own code body. —————>

outu eax, outs eax, outnl //outu is for unsigned values and outs is for signed. outnl is a new line

