A constant boolean condition is which no variables occur. For instance, (2<3) is a constant

boolean condition. Such conditions evaluate to the same truth value, irrespective of the values

of the variables in the program. An if statement whose boolean condition is constant, can be

replaced completely by either its then branch (if the boolean condition evaluates to true), or

by its else branch (if the boolean condition evaluates to false). Write a Prolog program that

simplifies toy programs in the manner described above.

Sample interaction:

?- P = (

a = 240 ; b = 144 ; i = 0 ;

while ( a \= b ) do {

if ( 2+3+(2>1) > 1\*1/1 )

then { b = b - a }

else { a = a - b } ;

i = i + 1 ;

} ), simplify\_if(P,NewP).

NewP = ( % unfortunately, Prolog won’t pretty-print output

a = 240 ; b = 144 ; i = 0 ;

while ( a \= b ) do {

{ b = b - a }

i = i + 1 ;

} )

P = ... % value of P will be printed as well, but we save space here