

1.

a. $z\ y$

b. $x\ y$

2.

a. y

b. x

c. $ayby$

d. $(\lambda z.z)y$

e. $(\lambda z.z)\ (y)\ (a)$

f. $(\lambda y.y\ x)\ (\lambda y.y)$

3.

$((\lambda x.x)\ \text{false}\ \text{true})\ ((\lambda x.x)\ \text{false}\ \text{true})\ \text{true}$

$((\lambda x.x)\ \text{false}\ \text{true})\ \text{true}\ \text{false}\ \text{true}$

$((\lambda x.x)\ \text{false}\ \text{true})\ (\lambda x.\lambda y.x)\ \text{false}\ \text{true}$

$((\lambda x.x)\ \text{false}\ \text{true})\ \text{false}$

$\text{false}\ \text{false}\ \text{true}$

$(\lambda x.\lambda y.y)\ \text{false}\ \text{true}$

true

// not too sure what the desired answer is... Just the first line?

1. Was accepted as hardware lacked floating-point support, so it had to be done with software.

2. Fortran 2 fixed many bugs in Fortran 1's compilation system. Also added independent compilation of subroutines, meaning small changes did not require complete recompilation.

3. Logical loop control, if-then-else

4. ACM and GAMM

5. ALGOL 60

6. Never widely used. Non-portable, wasn't supported by IBM.

7. Programs created with functions, instructions which create output from its input.

8. Free variables are variables used in a function and in which its value is dependent on the context the function is invoked from. Bounded variables' evaluation is not defined by the context of its calling.

9. Lambda expressions which cannot be reduced are referred to as a normal form.