```
entry:
                    %InputX.addr = alloca float, align 4
                    %sign = alloca i32, align 4
                    %OutputX = alloca float, align 4
                    %xInput = alloca float, align 4
                    %xNPrimeofX = alloca float, align 4
                    %expValues = alloca float, align 4
                    %xK2 = alloca float, align 4
                    %xK2_2 = alloca float, align 4
                    %xK2_3 = alloca float, align 4
                    %xK2_4 = alloca float, align 4
                    %xK2_5 = alloca float, align 4
                    %xLocal = alloca float, align 4
                    %xLocal_1 = alloca float, align 4
                    %xLocal_2 = alloca float, align 4
                    %xLocal_3 = alloca float, align 4
                    store float %InputX, float* %InputX.addr, align 4
                    %0 = load float* %InputX.addr, align 4
                    %conv = fpext float %0 to double
                    %cmp = fcmp olt double %conv, 0.000000e+00
                    br i1 %cmp, label %if.then, label %if.else
                              T
                                                        F
if.then:
%1 = load float* %InputX.addr, align 4
                                                  if.else:
% sub = fsub float -0.000000e+00, %1
                                                   store i32 0, i32* %sign, align 4
store float %sub, float* %InputX.addr, align 4
                                                   br label %if.end
store i32 1, i32* %sign, align 4
br label %if.end
               if.end:
               %2 = load float* %InputX.addr, align 4
               store float %2, float* %xInput, align 4
               %3 = load float* %InputX.addr, align 4
               %mul = fmul float -5.000000e-01, %3
               %4 = load float* %InputX.addr, align 4
               %mul1 = fmul float %mul, %4
               %conv2 = fpext float %mul1 to double
               %call = call double @exp(double %conv2) #1
               %conv3 = fptrunc double %call to float
               store float %conv3, float* %expValues, align 4
               %5 = load float* %expValues, align 4
               store float %5, float* %xNPrimeofX, align 4
               %6 = load float* %xNPrimeofX, align 4
               %conv4 = fpext float %6 to double
               %mul5 = fmul double %conv4, 0x3FD9884533D43651
               %conv6 = fptrunc double %mul5 to float
               store float %conv6, float* %xNPrimeofX, align 4
               %7 = load float* %xInput, align 4
               %conv7 = fpext float %7 to double
               %mul8 = fmul double 2.316419e-01, %conv7
               %conv9 = fptrunc double %mul8 to float
               store float %conv9, float* %xK2, align 4
               \%8 = \text{load float* } \%xK2, \text{ align } 4
               %conv10 = fpext float %8 to double
               %add = fadd double 1.000000e+00, %conv10
               %conv11 = fptrunc double %add to float
               store float %conv11, float* %xK2, align 4
               \%9 = \text{load float* } \%xK2, \text{ align } 4
               %conv12 = fpext float %9 to double
               %div = fdiv double 1.000000e+00, %conv12
               %conv13 = fptrunc double %div to float
               store float %conv13, float* %xK2, align 4
               %10 = load float* %xK2, align 4
               %11 = load float* %xK2, align 4
               %mul14 = fmul float %10, %11
               store float %mul14, float* %xK2 2, align 4
               %12 = load float* %xK2_2, align 4
               %13 = load float* %xK2, align 4
               %mul15 = fmul float %12, %13
               store float %mul15, float* %xK2_3, align 4
               %14 = load float* %xK2_3, align 4
               %15 = load float* %xK2, align 4
               %mul16 = fmul float %14, %15
               store float %mul16, float* %xK2_4, align 4
               \%16 = load float* \%xK2_4, align 4
               \%17 = load float* \%xK2, align 4
               %mul17 = fmul float %16, %17
               store float %mul17, float* %xK2_5, align 4
               %18 = load float* %xK2, align 4
               %conv18 = fpext float %18 to double
               %mul19 = fmul double %conv18, 0x3FD470BF3A92F8EC
               %conv20 = fptrunc double %mul19 to float
               store float %conv20, float* %xLocal_1, align 4
               %19 = load float* %xK2_2, align 4
               %conv21 = fpext float %19 to double
               %mul22 = fmul double %conv21, 0xBFD6D1F0E5A8325B
               %conv23 = fptrunc double %mul22 to float
               store float %conv23, float* %xLocal_2, align 4
               \%20 = load float* \%xK2_3, align 4
               %conv24 = fpext float %20 to double
               %mul25 = fmul double %conv24, 0x3FFC80EF025F5E68
               %conv26 = fptrunc double %mul25 to float
               store float %conv26, float* %xLocal_3, align 4
               %21 = load float* %xLocal_2, align 4
               %22 = load float* %xLocal_3, align 4
               %add27 = fadd float %21, %22
               store float %add27, float* %xLocal 2, align 4
               %23 = load float* %xK2 4, align 4
               %conv28 = fpext float %23 to double
               %mul29 = fmul double %conv28, 0xBFFD23DD4EF278D0
               %conv30 = fptrunc double %mul29 to float
               store float %conv30, float* %xLocal_3, align 4
               %24 = load float* %xLocal_2, align 4
               %25 = load float* %xLocal 3, align 4
               %add31 = fadd float %24, %25
               store float %add31, float* %xLocal_2, align 4
               \%26 = load float* \%xK2_5, align 4
               %conv32 = fpext float %26 to double
               %mul33 = fmul double %conv32, 0x3FF548CDD6F42943
               %conv34 = fptrunc double %mul33 to float
               store float %conv34, float* %xLocal_3, align 4
                \%27 = load float* \%xLocal_2, align 4
               %28 = load float* %xLocal_3, align 4
               %add35 = fadd float %27, %28
               store float %add35, float* %xLocal_2, align 4
               %29 = load float* %xLocal_2, align 4
               %30 = load float* %xLocal 1, align 4
               %add36 = fadd float %29, %30
               store float %add36, float* %xLocal_1, align 4
               %31 = load float* %xLocal_1, align 4
               %32 = load float* %xNPrimeofX, align 4
               %mul37 = fmul float %31, %32
               store float %mul37, float* %xLocal, align 4
               %33 = load float* %xLocal, align 4
               %conv38 = fpext float %33 to double
               %sub39 = fsub double 1.000000e+00, %conv38
               %conv40 = fptrunc double %sub39 to float
               store float %conv40, float* %xLocal, align 4
               %34 = load float* %xLocal, align 4
               store float %34, float* %OutputX, align 4
               %35 = load i32* %sign, align 4
               %tobool = icmp ne i32 %35, 0
               br i1 %tobool, label %if.then41, label %if.end45
                            \mathbf{T}
                                                           F
     if.then41:
      %36 = load float* %OutputX, align 4
      %conv42 = fpext float %36 to double
      %sub43 = fsub double 1.000000e+00, %conv42
      %conv44 = fptrunc double %sub43 to float
      store float %conv44, float* %OutputX, align 4
      br label %if.end45
```

%37 = load float* %OutputX, align 4 ret float %37

CFG for '_Z4CNDFf' function

if.end45: