%strike.addr = alloca float, align 4 %rate.addr = alloca float, align 4 %volatility.addr = alloca float, align 4 %time.addr = alloca float, align 4 %otype.addr = alloca i32, align 4 %timet.addr = alloca float, align 4 %N1.addr = alloca float*, align 8 %N2.addr = alloca float*, align 8 %OptionPrice = alloca float, align 4 %xRiskFreeRate = alloca float, align 4 %xVolatility = alloca float, align 4 %xTime = alloca float, align 4 %xSqrtTime = alloca float, align 4 %logValues = alloca float, align 4 %xLogTerm = alloca float, align 4 %xD1 = alloca float, align 4 %xD2 = alloca float, align 4%xPowerTerm = alloca float, align 4 %xDen = alloca float, align 4 %d1 = alloca float, align 4 %d2 = alloca float, align 4%FutureValueX = alloca float, align 4 %NofXd1 = alloca float, align 4 %NofXd2 = alloca float, align 4 %NegNofXd1 = alloca float, align 4 %NegNofXd2 = alloca float, align 4 store float %sptprice, float* %sptprice.addr, align 4 store float %strike, float* %strike.addr, align 4 store float %rate, float* %rate.addr, align 4 store float %volatility, float* %volatility.addr, align 4 store float %time, float* %time.addr, align 4 store i32 %otype, i32* %otype.addr, align 4 store float %timet, float* %timet.addr, align 4 store float* %N1, float** %N1.addr, align 8 store float* %N2, float** %N2.addr, align 8 %0 = load float* %rate.addr, align 4 store float %0, float* %xRiskFreeRate, align 4 %1 = load float* %volatility.addr, align 4 store float %1, float* %xVolatility, align 4 %2 = load float* %time.addr, align 4 store float %2, float* %xTime, align 4 %3 = load float* %xTime, align 4 %conv = fpext float %3 to double %call = call double @sqrt(double %conv) #1 %conv1 = fptrunc double %call to float store float %conv1, float* %xSqrtTime, align 4 %4 = load float* %sptprice.addr, align 4 %5 = load float* %strike.addr, align 4 %div = fdiv float %4, %5 %conv2 = fpext float %div to double %call3 = call double @log(double %conv2) #1 %conv4 = fptrunc double %call3 to float store float %conv4, float* %logValues, align 4 %6 = load float* %logValues, align 4 store float %6, float* %xLogTerm, align 4 %7 = load float* %xVolatility, align 4 %8 = load float* %xVolatility, align 4 %mul = fmul float %7, %8 store float %mul, float* %xPowerTerm, align 4 %9 = load float* %xPowerTerm, align 4 %conv5 = fpext float %9 to double %mul6 = fmul double %conv5, 5.000000e-01 %conv7 = fptrunc double %mul6 to float store float %conv7, float* %xPowerTerm, align 4 %10 = load float* %xRiskFreeRate, align 4 %11 = load float* %xPowerTerm, align 4 %add = fadd float %10, %11 store float %add, float* %xD1, align 4 %12 = load float* %xD1, align 4%13 = load float* %xTime, align 4 %mul8 = fmul float %12, %13 store float %mul8, float* %xD1, align 4 %14 = load float* %xD1, align 4%15 = load float* %xLogTerm, align 4 %add9 = fadd float %14, %15 store float %add9, float* %xD1, align 4 %16 = load float* %xVolatility, align 4 %17 = load float* %xSqrtTime, align 4 %mul10 = fmul float %16, %17 store float %mul10, float* %xDen, align 4 %18 = load float* %xD1, align 4%19 = load float* %xDen, align 4 %div11 = fdiv float %18, %19 store float %div11, float* %xD1, align 4 %20 = load float* %xD1, align 4%21 = load float* %xDen, align 4 %sub = fsub float %20, %21 store float %sub, float* %xD2, align 4 %22 = load float* %xD1, align 4store float %22, float* %d1, align 4 %23 = load float* %xD2, align 4store float %23, float* %d2, align 4 %24 = load float* %d1, align 4%call12 = call float @_Z4CNDFf(float %24) store float %call12, float* %NofXd1, align 4 %25 = load float* %NofXd1, align 4 %conv13 = fpext float %25 to double %cmp = fcmp ogt double %conv13, 1.000000e+00 br i1 %cmp, label %if.then, label %if.end T F if.then: %call14 = call %"class.std::basic_ostream"* @_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc(%"class.std::basic_ostream"* @_ZSt4cerr, i8* getelementptr inbounds ([18 x i8]* @.str, i32 0, i32 0)) %call15 = call %"class.std::basic_ostream"* @_ZNSolsEPFRSoS_E(%"class.std::basic_ostream"* %call14, %"class.std::basic_ostream"*)* @_ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_) br label %if.end if.end: %26 = load float* %d2, align 4%call16 = call float @_Z4CNDFf(float %26) store float %call16, float* %NofXd2, align 4 %27 = load float* %NofXd2, align 4 %conv17 = fpext float %27 to double %cmp18 = fcmp ogt double %conv17, 1.000000e+00 br i1 %cmp18, label %if.then19, label %if.end22 F if.then19: %call20 = call %"class.std::basic_ostream"* @_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc(%"class.std::basic_ostream"* @_ZSt4cerr, i8* getelementptr inbounds ([18 x i8]* @.str, i32 0, i32 0)) %call21 = call %"class.std::basic ostream"* @ ZNSolsEPFRSoS E(%"class.std::basic ostream"* %call20, %"class.std::basic ostream"*)* @ ZSt4endIIcSt11char traitsIcEERSt13basic ostreamIT TO ES6 br label %if.end22 if.end22: %28 = load float* %NofXd1, align 4%29 = load float** %N1.addr, align 8 store float %28, float* %29, align 4 %30 = load float* %NofXd2, align 4 %31 = load float** %N2.addr, align 8 store float %30, float* %31, align 4 %32 = load float* %strike.addr, align 4 %conv23 = fpext float %32 to double %33 = load float* %rate.addr, align 4 % sub24 = fsub float -0.000000e+00, % 33 %34 = load float* %time.addr, align 4 %mul25 = fmul float %sub24, %34 %conv26 = fpext float %mul25 to double %call27 = call double @exp(double %conv26) #1 %mul28 = fmul double %conv23, %call27 %conv29 = fptrunc double %mul28 to float store float %conv29, float* %FutureValueX, align 4 %35 = load i32* %otype.addr, align 4 %cmp30 = icmp eq i32 %35, 0 br i1 %cmp30, label %if.then31, label %if.else if.else: %40 = load float* %NofXd1, align 4 %conv35 = fpext float %40 to double %sub36 = fsub double 1.000000e+00, %conv35 %conv37 = fptrunc double %sub36 to float if.then31: store float %conv37, float* %NegNofXd1, align 4 %36 = load float* %sptprice.addr, align 4 %41 = load float* %NofXd2, align 4 %37 = load float* %NofXd1, align 4 %conv38 = fpext float %41 to double %mul32 = fmul float %36, %37 %sub39 = fsub double 1.000000e+00, %conv38 %38 = load float* %FutureValueX, align 4 %conv40 = fptrunc double %sub39 to float %39 = load float* %NofXd2, align 4 store float %conv40, float* %NegNofXd2, align 4 %mul33 = fmul float %38, %39 %42 = load float* %FutureValueX, align 4 %sub34 = fsub float %mul32, %mul33 %43 = load float* %NegNofXd2, align 4 store float %sub34, float* %OptionPrice, align 4 %mul41 = fmul float %42, %43 br label %if.end44 %44 = load float* %sptprice.addr, align 4 %45 = load float* %NegNofXd1, align 4 %mul42 = fmul float %44, %45 %sub43 = fsub float %mul41, %mul42 store float %sub43, float* %OptionPrice, align 4 br label %if.end44 if.end44: %46 = load float* %OptionPrice, align 4 ret float %46

entry:

%sptprice.addr = alloca float, align 4