```
%5:
                                                           %6 = tail call i64 @ Z13get global idj(i32 0) #3
                                                           \%7 = \text{trunc } i64 \%6 \text{ to } i32
                                                           %8 = icmp slt i32 \%7, \%3
                                                          br i1 %8, label %9, label %53
                                                                                                F
                                    %9:
                                    9:
                                     %10 = \text{shl } i64 \%6, 32
                                     %11 = ashr exact i64 %10, 32
                                     %12 = getelementptr inbounds float, float* %2, i64 %11
                                     store float 0.000000e+00, float* %12, align 4, !tbaa !10
                                     %13 = icmp sgt i32 %4, 0
                                     br i1 %13, label %14, label %53
                                                  Т
                                                                                 F
                        %14:
                        14:
                        %15 = \text{mul nsw i} 32 \%7, \%4
                         %16 = \text{sext i} 32 \% 15 \text{ to i} 64
                         %17 = \text{zext i} 32 \% 4 \text{ to i} 64
                         %18 = and i64 \%17, 1
                         %19 = icmp eq i32 %4, 1
                        br i1 %19, label %42, label %20
                               Τ
                                  %20:
                                  20:
                                   %21 = sub nsw i64 %17, %18
                                   br label %22
            %22:
             %23 = \text{phi float} [ 0.0000000e+00, %20 ], [ %38, %22 ]
             %24 = phi i64 [ 0, %20 ], [ %39, %22 ]
             %25 = phi i64 [ %21, %20 ], [ %40, %22 ]
             %26 = add nsw i64 %24, %16
             %27 = getelementptr inbounds float, float* %0, i64 %26
             %28 = load float, float* %27, align 4, !tbaa !10
             %29 = getelementptr inbounds float, float* %1, i64 %24
             %30 = load float, float* %29, align 4, !tbaa !10
             %31 = tail call float @llvm.fmuladd.f32(float %28, float %30, float %23)
             store float %31, float* %12, align 4, !tbaa !10
             %32 = \text{ or } i64 \%24, 1
             %33 = add nsw i64 %32, %16
             %34 = getelementptr inbounds float, float* %0, i64 %33
             %35 = load float, float* %34, align 4, !tbaa !10
             %36 = getelementptr inbounds float, float* %1, i64 %32
             %37 = load float, float* %36, align 4, !tbaa !10
             %38 = tail call float @llvm.fmuladd.f32(float %35, float %37, float %31)
             store float %38, float* %12, align 4, !tbaa !10
             %39 = add nuw nsw i64 %24, 2
             %40 = add i64 \%25, -2
             %41 = icmp \ eq \ i64 \ \%40, \ 0
             br i1 %41, label %42, label %22
                                                                    F
%42:
42:
%43 = \text{phi float} [0.0000000e+00, %14], [%38, %22]
%44 = phi i64 [ 0, %14 ], [ %39, %22 ]
%45 = icmp eq i64 \%18, 0
br i1 %45, label %53, label %46
                                         F
               %46:
               46:
                %47 = add nsw i64 %44, %16
               %48 = getelementptr inbounds float, float* %0, i64 %47
               %49 = load float, float* %48, align 4, !tbaa !10
               %50 = getelementptr inbounds float, float* %1, i64 %44
               %51 = load float, float* %50, align 4, !tbaa !10
                %52 = tail call float @llvm.fmuladd.f32(float %49, float %51, float %43)
               store float %52, float* %12, align 4, !tbaa !10
               br label %53
                                                                      %53:
                                                                     53:
                                                                      ret void
```