```
%call = tail call i32 @initialise_arrays(i8* getelementptr inbounds ([5 x
                                                                                   ... i8], [5 x i8]* @ func .s271, i64 0, i64 0)) #6
                                                                                   %t1 = getelementptr inbounds %struct.args t, %struct.args_t* %func_args, i64
                                                                                   ... 0, i32 0
                                                                                   %call1 = tail call i32 @gettimeofday(%struct.timeval* %t1, i8* null) #6
                                                                                   br label %for.cond2.preheader
                                                                                            for.cond2.preheader:
                                                                                            %nl.029 = phi i32 [ 0, %entry ], [ %inc15, %for.cond.cleanup4 ]
                                                                                            br label %for.body5
                                                             for.body5:
                                                              %indvars.iv = phi i64 [ 0, %for.cond2.preheader ], [ %indvars.iv.next.7,
                                                              ... %for.inc.7 ]
                                                              %arrayidx = getelementptr inbounds [32000 x float], [32000 x float]* @b, i64
                                                              .. 0, i64 %indvars.iv
                                                              %0 = load float, float* %arrayidx, align 32, !tbaa !4
                                                              %cmp6 = fcmp ogt float %0, 0.000000e+00
                                                              br i1 %cmp6, label %if.then, label %for.inc
                                                                                                                          F
                                        if.then:
                                         %arrayidx10 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
                                         ... i64 0, i64 %indvars.iv
                                         %1 = load float, float* %arrayidx10, align 32, !tbaa !4
                                         %mul = fmul float %0, %1
                                         %arrayidx12 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
                                         ... i64 0, i64 %indvars.iv
                                         %2 = load float, float* %arrayidx12, align 32, !tbaa !4
                                         %add = fadd float %2, %mul
                                         store float %add, float* %arrayidx12, align 32, !tbaa !4
                                         br label %for.inc
                                         for.inc:
                                         %indvars.iv.next = or i64 %indvars.iv, 1
                                         %arrayidx.1 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                                         ... i64 0, i64 %indvars.iv.next
                                         %3 = load float, float* %arrayidx.1, align 4, !tbaa !4
                                         %cmp6.1 = fcmp ogt float %3, 0.000000e+00
                                         br i1 %cmp6.1, label %if.then.1, label %for.inc.1
                                                                                                    F
                 %arrayidx10.1 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
                ... i64 0, i64 %indvars.iv.next
                 %4 = load float, float* %arrayidx10.1, align 4, !tbaa !4
                 %mul.1 = fmul float %3, %4
                 %arrayidx12.1 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
                ... i64 Ő, i64 %indvars.iv.next
                %5 = load float, float* %arrayidx12.1, align 4, !tbaa !4 %add.1 = fadd float %5, %mul.1
                store float %add.1, float* %arrayidx12.1, align 4, !tbaa !4
                 br label %for.inc.1
                             for.inc.1:
                              %indvars.iv.next.1 = or i64 %indvars.iv, 2
                             %arrayidx.2 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                             ... i64 Ő, i64 %indvars.iv.next.1
                             %6 = load float, float* %arrayidx.2, align 8, !tbaa !4 %cmp6.2 = fcmp ogt float %6, 0.000000e+00
                             br i1 %cmp6.2, label %if.then.2, label %for.inc.2
     if.then.2:
     %arrayidx10.2 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
     ... i64 0, i64 %indvars.iv.next.1
     %7 = load float, float* %arrayidx10.2, align 8, !tbaa !4 %mul.2 = fmul float %6, %7
     %arrayidx12.2 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
     ... i64 0, i64 %indvars.iv.next.1
     %8 = load float, float* %arrayidx12.2, align 8, !tbaa !4
     %add.2 = fadd float %8, %mul.2
     store float %add.2, float* %arrayidx12.2, align 8, !tbaa !4 br label %for.inc.2
                        for.inc.2:
                         %indvars.iv.next.2 = or i64 %indvars.iv, 3
                         %arrayidx.3 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                        ... i64 0, i64 %indvars.iv.next.2
                        %9 = load float, float* %arrayidx.3, align 4, !tbaa !4
                        %cmp6.3 = fcmp ogt float %9, 0.000000e+00
br i1 %cmp6.3, label %if.then.3, label %for.inc.3
if.then.3:
%arrayidx10.3 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
... i64 0, i64 %indvars.iv.next.2
%10 = load float, float* %arrayidx10.3, align 4, !tbaa !4
%mul.3 = fmul float %9, %10
%arrayidx12.3 = getelementptr inbounds [32000 x float], [32000 x float]* @a, ... i64 0, i64 %indvars.iv.next.2
%11 = load float, float* %arrayidx12.3, align 4, !tbaa !4
%add.3 = fadd float %11, %mul.3
store float %add.3, float* %arrayidx12.3, align 4, !tbaa !4
br label %for.inc.3
                               for.inc.3:
                               %indvars.iv.next.3 = or i64 %indvars.iv, 4
                               %arrayidx.4 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                               ... i64 0, i64 %indvars.iv.next.3
                               %12 = load float, float* %arrayidx.4, align 16, !tbaa !4
                               %cmp6.4 = fcmp ogt float %12, 0.000000e+00
                               br i1 %cmp6.4, label %if.then.4, label %for.inc.4
                                                                                          F
      if.then.4:
      %arrayidx10.4 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
      ... i64 0, i64 %indvars.iv.next.3
      %13 = load float, float* %arrayidx10.4, align 16, !tbaa !4
      %mul.4 = fmul float %12, %13
      %arrayidx12.4 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
      ... i64 0, i64 %indvars.iv.next.3
      %14 = load float, float* %arrayidx12.4, align 16, !tbaa !4
      %add.4 = fadd float %14, %mul.4
      store float %add.4, float* %arrayidx12.4, align 16, !tbaa !4
       br label %for.inc.4
                              for.inc.4:
                               %indvars.iv.next.4 = or i64 %indvars.iv, 5
                               %arrayidx.5 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                               .. i64 0, i64 %indvars.iv.next.4
                               %15 = load float, float* %arrayidx.5, align 4, !tbaa !4
                               %cmp6.5 = fcmp ogt float %15, 0.000000e+00
                               br i1 %cmp6.5, label %if.then.5, label %for.inc.5
                                                                                          F
      if.then.5:
      %arrayidx10.5 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
      ... i64 0, i64 %indvars.iv.next.4
      %16 = load float, float* %arrayidx10.5, align 4, !tbaa !4
      %mul.5 = fmul float %15, %16
      %arrayidx12.5 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
      ... i64 Ŏ, i64 %indvars.iv.next.4
      %17 = load float, float* %arrayidx12.5, align 4, !tbaa !4
      %add.5 = fadd float %17, %mul.5
      store float %add.5, float* %arrayidx12.5, align 4, !tbaa !4
       br label %for.inc.5
                              for.inc.5:
                               %indvars.iv.next.5 = or i64 %indvars.iv, 6
                               %arrayidx.6 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                               ... i64 0, i64 %indvars.iv.next.5
                               %18 = load float, float* %arrayidx.6, align 8, !tbaa !4
                               %cmp6.6 = fcmp ogt float %18, 0.000000e+00
                               br i1 %cmp6.6, label %if.then.6, label %for.inc.6
      if.then.6:
      %arrayidx10.6 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
      ... i64 0, i64 %indvars.iv.next.5
      %19 = load float, float* %arrayidx10.6, align 8, !tbaa !4
      %mul.6 = fmul float %18, %19
      %arrayidx12.6 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
      ... i64 0, i64 %indvars.iv.next.5
      %20 = load float, float* %arrayidx12.6, align 8, !tbaa !4
      %add.6 = fadd float %20, %mul.6
      store float %add.6, float* %arrayidx12.6, align 8, !tbaa !4
       br label %for.inc.6
                                         for.inc.6:
                                         %indvars.iv.next.6 = or i64 %indvars.iv, 7
                                         %arrayidx.7 = getelementptr inbounds [32000 x float], [32000 x float]* @b,
                                         ... i64 0, i64 %indvars.iv.next.6
                                         %21 = load float, float* %arrayidx.7, align 4, !tbaa !4 %cmp6.7 = fcmp ogt float %21, 0.000000e+00
                                         br i1 %cmp6.7, label %if.then.7, label %for.inc.7
                                                                                                    F
                   if.then.7:
                    %arrayidx10.7 = getelementptr inbounds [32000 x float], [32000 x float]* @c,
                   ... i64 0, i64 %indvars.iv.next.6
                    %22 = load float, float* %arrayidx10.7, align 4, !tbaa !4
                    %mul.7 = fmul float %21, %22
                    %arrayidx12.7 = getelementptr inbounds [32000 x float], [32000 x float]* @a,
                    ... i64 0, i64 %indvars.iv.next.6
                    %23 = load float, float* %arrayidx12.7, align 4, !tbaa !4
                    %add.7 = fadd float %23, %mul.7
                    store float %add.7, float* %arrayidx12.7, align 4, !tbaa !4
                    br label %for.inc.7
                                                                         for.inc.7:
                                                                         %indvars.iv.next.7 = add nuw nsw i64 %indvars.iv, 8
                                                                         %exitcond.7.not = icmp eq i64 %indvars.iv.next.7, 32000
                                                                         br i1 %exitcond.7.not, label %for.cond.cleanup4, label %for.body5,
                                                                         ... !llvm.loop !8
                                                                    for.cond.cleanup4:
                                                                    %call13 = tail call i32 @dummy(float* getelementptr inbounds ([32000 x
                                                                    ... float], [32000 x float]* @a, i64 0, i64 0), float* getelementptr inbounds
                                                                    ... ([32000 x float], [32000 x float]* @b, i64 0, i64 0), float* getelementptr
                                                                    ... inbounds ([32000 x float], [32000 x float]* @c, i64 0, i64 0), float*
                                                                    ... getelementptr inbounds ([32000 x float], [32000 x float]* @d, i64 0, i64 0),
                                                                    ... float* getelementptr inbounds ([32000 x float], [32000 x float]* @e, i64 0,
                                                                    ... i64 0), [256 x float]* getelementptr inbounds ([256 x [256 x float]], [256 x
                                                                    ... [256 x float]]* @aa, i64 0, i64 0), [256 x float]* getelementptr inbounds
                                                                    ... ([256 x [256 x float]], [256 x [256 x float]]* @bb, i64 0, i64 0), [256 x
                                                                    ... float]* getelementptr inbounds ([256 x [256 x float]], [256 x [256 x float]]*
                                                                    ... @cc, i64 0, i64 0), float 0.000000e+00) #6
                                                                    %inc15 = add nuw nsw i32 %nl.029, 1
                                                                    %exitcond.not = icmp eq i32 %inc15, 400000
                                                                    br i1 %exitcond.not, label %for.cond.cleanup, label %for.cond2.preheader,
                                                                    ...!llvm.loop!2
                                                                                                                                F
                                                                                       Τ
                                             for.cond.cleanup:
                                              %t2 = getelementptr inbounds %struct.args t, %struct.args t* %func args, i64
                                             ... 0, i32 1
                                              %call17 = tail call i32 @gettimeofday(%struct.timeval* nonnull %t2, i8*
                                             ... null) #6
```

entry:

CFG for 's271' function

... i8], [5 x i8]* @ func .s271, i64 0, i64 0)) #6

ret float %call18

%call18 = tail call float @calc checksum(i8* getelementptr inbounds ([5 x