```
%branch.addr = alloca i32, align 4
%i.addr = alloca i32, align 4
%A = alloca [10 x i32], align 16
%k = alloca i32, align 4
%result = alloca i32, align 4
store i32 %branch, i32* %branch.addr, align 4
call void @llvm.dbg.declare(metadata i32* %branch.addr, metadata !289,
.. metadata !DIExpression()), !dbg !290
%branch.addr1 = bitcast i32* %branch.addr to i8*
call void @llvm.var.annotation(i8* %branch.addr1, i8* getelementptr inbounds
... ([7 x i8], [7 x i8]* @.str, i32 0, i32 0), i8* getelementptr inbounds ([21 x
... i8], [21 x i8]* @.str.1, i32 0, i32 0), i32 8, i8* null)
store i32 %i, i32* %i.addr, align 4
call void @llvm.dbg.declare(metadata i32* %i.addr, metadata !291, metadata
...!DIExpression()),!dbg!292
call void @llvm.dbg.declare(metadata [10 x i32]* %A, metadata !293, metadata
... !DIExpression()), !dbg !297
\%0 = \text{bitcast} [10 \times i32]^* \% \text{A to } i8^*, !dbg !297
call void @llvm.memcpy.p0i8.p0i8.i64(i8* align 16 %0, i8* align 16 bitcast
... ([10 x i32]* @ const. Z12targetBranchii.A to i8*), i64 40, i1 false), !dbg
call void @llvm.dbg.declare(metadata i32* %k, metadata !298, metadata
...!DIExpression()), !dbg!299
%1 = load i32, i32* %branch.addr, align 4, !dbg !300
%idxprom = sext i32 %1 to i64, !dbg !301
%arrayidx = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
... %idxprom, !dbg !301
\%2 = load i32, i32* \%arrayidx, align 4, !dbg !301
store i32 %2, i32* %k, align 4, !dbg !299
call void @llvm.dbg.declare(metadata i32* %result, metadata !302, metadata
...!DIExpression()),!dbg!303
store i32 0, i32* %result, align 4, !dbg !303
%3 = load i32, i32* %k, align 4, !dbg !304
switch i32 %3, label %sw.default [
i32 0, label %sw.bb
i32 1, label %sw.bb5
i32 2, label %sw.bb21
], !dbg !305, !prof !306
        def
```

```
sw.default:
%175 = alloca i32, align 4, !dbg !355
%176 = load i32, i32* %175, align 4, !dbg !355
%177 = alloca i32, align 4, !dbg !355
%178 = load i32, i32* %177, align 4, !dbg !355
%179 = alloca i32, align 4, !dbg !355
%180 = load i32, i32* %179, align 4, !dbg !355
%181 = alloca i32, align 4, !dbg !355
 %182 = load i32, i32* %181, align 4, !dbg !355
%183 = alloca i32, align 4, !dbg !355
%184 = load i32, i32* %183, align 4, !dbg !355
%185 = alloca i32, align 4, !dbg !355
%186 = load i32, i32* %185, align 4, !dbg !355
%187 = alloca i32, align 4, !dbg !355
%188 = load i32, i32* %187, align 4, !dbg !355
%189 = alloca i32, align 4, !dbg !355
 %190 = load i32, i32* %189, align 4, !dbg !355
%191 = alloca i32, align 4, !dbg !355
%192 = load i32, i32* %191, align 4, !dbg !355
%193 = alloca i32, align 4, !dbg !355
%194 = load i32, i32* %193, align 4, !dbg !355
%195 = alloca i32, align 4, !dbg !355
%196 = load i32, i32* %195, align 4, !dbg !355
%197 = alloca i32, align 4, !dbg !355
%198 = load i32, i32* %197, align 4, !dbg !355
%199 = alloca i32, align 4, !dbg !355
%200 = load i32, i32* %199, align 4, !dbg !355
%201 = alloca i32, align 4, !dbg !355
%202 = load i32, i32* %201, align 4, !dbg !355
 %203 = mul i32 %202, 42, !dbg !355
 %204 = mul i32 %203, 42, !dbg !355
%205 = mul i32 %204, 42, !dbg !355
%206 = mul i32 %205, 42, !dbg !355
 %207 = mul i32 %206, 42, !dbg !355
 %208 = add i32 %207, 42, !dbg !355
 %209 = add i32 %208, 42, !dbg !355
 %210 = add i32 %209, 42, !dbg !355
 %211 = add i32 %210, 42, !dbg !355
 %212 = add i32 %211, 42, !dbg !355
%213 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %213, align 4, !dbg !355
%214 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %214, align 4, !dbg !355
%215 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %215, align 4, !dbg !355
%216 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %216, align 4, !dbg !355
%217 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %217, align 4, !dbg !355
%218 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %218, align 4, !dbg !355
%219 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %219, align 4, !dbg !355
 %220 = alloca i32, align 4, !dbg !355
 store volatile i32 42, i32* %220, align 4, !dbg !355
%221 = alloca i32, align 4, !dbg !355
store volatile i32 42, i32* %221, align 4, !dbg !355
%pgocount3 = load i64, i64* getelementptr inbounds ([4 x i64], [4 x i64]* ... @__profc__Z12targetBranchii, i64 0, i64 3), align 8, !dbg !355
%22\overline{2} = add i64 \% pgocount3, 1, !dbg !355
store i64 %222, i64* getelementptr inbounds ([4 x i64], [4 x i64]* ... @_profc_Z12targetBranchii, i64 0, i64 3), align 8, !dbg !355
store i32 -1, i32* %result, align 4, !dbg !355
br label %sw.epilog, !dbg !356
```

```
%4 = alloca i32, align 4, !dbg !307
 %5 = load i32, i32* %4, align 4, !dbg !307
 %6 = alloca i32. align 4. !dbg !307
 %7 = load i32, i32* %6, align 4, !dbg !307
%8 = alloca i32, align 4, !dbg !307
%9 = load i32, i32* %8, align 4, !dbg !307
 %10 = alloca i32, align 4, !dbg !307
 %11 = load i32, i32* %10, align 4, !dbg !307
 %12 = alloca i32, align 4, !dbg !307
 %13 = load i32, i32* %12, align 4, !dbg !307
 %14 = alloca i32, align 4, !dbg !307
 %15 = load i32, i32* %14, align 4, !dbg !307
 %16 = alloca i32, align 4, !dbg !307
%17 = load i32, i32* %16, align 4, !dbg !307
%18 = alloca i32, align 4, !dbg !307
 %19 = load i32, i32* %18, align 4, !dbg !307
 %20 = alloca i32, align 4, !dbg !307
 %21 = load i32, i32* %20, align 4, !dbg !307
 %22 = alloca i32, align 4, !dbg !307
 %23 = load i32, i32* %22, align 4, !dbg !307
 %24 = alloca i32, align 4, !dbg !307
 %25 = load i32, i32* %24, align 4, !dbg !307
 %26 = alloca i32, align 4, !dbg !307
 %27 = load i32, i32* %26, align 4, !dbg !307
 %28 = alloca i32, align 4, !dbg !307
 %29 = load i32, i32* %28, align 4, !dbg !307
 %30 = alloca i32, align 4, !dbg !307
 %31 = load i32, i32* %30, align 4, !dbg !307
 %32 = mul i32 %31, 42, !dbg !307
 %33 = mul i32 %32, 42, !dbg !307
 %34 = mul i32 %33, 42, !dbg !307
%35 = mul i32 %34, 42, !dbg !307
 %36 = mul i32 %35, 42, !dbg !307
 %37 = add i32 %36, 42, !dbg !307
 %38 = add i32 %37, 42, !dbg !307
 %39 = add i32 %38, 42, !dbg !307
 %40 = add i32 %39, 42, !dbg !307
 %41 = add i32 %40, 42, !dbg !307
 %42 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %42, align 4, !dbg !307
 %43 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %43, align 4, !dbg !307
 %44 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %44, align 4, !dbg !307
 %45 = alloca i32, align 4, !dbg !307
store volatile i32 42, i32* %45, align 4, !dbg !307
 %46 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %46, align 4, !dbg !307
 %47 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %47, align 4, !dbg !307
 %48 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %48, align 4, !dbg !307
 %49 = alloca i32, align 4, !dbg !307
store volatile i32 42, i32* %49, align 4, !dbg !307
 %50 = alloca i32, align 4, !dbg !307
 store volatile i32 42, i32* %50, align 4, !dbg !307
%pgocount = load i64, i64* getelementptr inbounds ([4 x i64], [4 x i64]*
... @__profc__Z12targetBranchii, i64 0, i64 2), align 8, !dbg !307
%51 = add i64 %pgocount, 1, !dbg !307
store i64 %51, i64* getelementptr inbounds ([4 x i64], [4 x i64]*
... @__profc__Z12targetBranchii, i64 0, i64 2), align 8, !dbg !307
%52 = load i32, i32* %result, align 4, !dbg !307
 %div = sdiv i32 %52, 3, !dbg !307
 store i32 %div, i32* %result, align 4, !dbg !307
 %53 = load i32, i32* %i.addr, align 4, !dbg !309
 %rem = srem i32 %53, 14, !dbg !310
 %mul = mul nsw i32 %rem, 243, !dbg !311
 %54 = load i32, i32* %result, align 4, !dbg !312
%xor = xor i32 %54, %mul, !dbg !312
 store i32 %xor, i32* %result, align 4, !dbg !312
 %55 = load i32, i32* %i.addr, align 4, !dbg !313
 %rem2 = srem i32 %55, 10, !dbg !314
 %idxprom3 = sext i32 %rem2 to i64, !dbg !315
 %arrayidx4 = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
  . %idxprom3, !dbg !315
 %56 = load i32, i32* %arrayidx4, align 4, !dbg !315
 %57 = load i32, i32* %result, align 4, !dbg !316
 %add = add nsw i32 %57, %56, !dbg !316
 store i32 %add, i32* %result, align 4, !dbg !316
```

br label %sw.epilog, !dbg !317

```
sw.bb5:
%58 = alloca i32, align 4, !dbg !318
%59 = load i32, i32* %58, align 4, !dbg !318
%60 = alloca i32, align 4, !dbg !318
%61 = load i32, i32* %60, align 4, !dbg !318
%62 = alloca i32, align 4, !dbg !318
%63 = load i32, i32* %62, align 4, !dbg !318
%64 = alloca i32, align 4, !dbg !318
%65 = load i32, i32* %64, align 4, !dbg !318
%66 = alloca i32, align 4, !dbg !318
%67 = load i32, i32* %66, align 4, !dbg !318
 %68 = alloca i32, align 4, !dbg !318
 %69 = load i32, i32* %68, align 4, !dbg !318
%70 = alloca i32, align 4, !dbg !318
%71 = load i32, i32* %70, align 4, !dbg !318
%72 = alloca i32, align 4, !dbg !318
%73 = load i32, i32* %72, align 4, !dbg !318
%74 = alloca i32, align 4, !dbg !318
%75 = load i32, i32* %74, align 4, !dbg !318
%76 = alloca i32, align 4, !dbg !318
%77 = load i32, i32* %76, align 4, !dbg !318
%78 = alloca i32, align 4, !dbg !318
%79 = load i32, i32* %78, align 4, !dbg !318
%80 = alloca i32, align 4, !dbg !318
%81 = load i32, i32* %80, align 4, !dbg !318
%82 = alloca i32, align 4, !dbg !318
%83 = load i32, i32* %82, align 4, !dbg !318
%84 = alloca i32, align 4, !dbg !318

%85 = load i32, i32* %84, align 4, !dbg !318
%86 = mul i32 %85, 42, !dbg !318
%87 = mul i32 %86, 42, !dbg !318
%88 = mul i32 %87, 42, !dbg !318
%89 = mul i32 %88, 42, !dbg !318
%90 = mul i32 %89, 42, !dbg !318
%91 = add i32 %90, 42, !dbg !318
%92 = add i32 %91, 42, !dbg !318
%93 = add i32 %92, 42, !dbg !318
%94 = add i32 %93, 42, !dbg !318
%95 = add i32 %94, 42, !dbg !318
%96 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %96, align 4, !dbg !318
%97 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %97, align 4, !dbg !318
%98 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %98, align 4, !dbg !318
%99 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %99, align 4, !dbg !318
%100 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %100, align 4, !dbg !318
%101 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %101, align 4, !dbg !318
%102 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %102, align 4, !dbg !318
%103 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %103, align 4, !dbg !318
%104 = alloca i32, align 4, !dbg !318
store volatile i32 42, i32* %104, align 4, !dbg !318
%pgocount1 = load i64, i64* getelementptr inbounds ([4 x i64], [4 x i64]*
... @_profc_Z12targetBranchii, i64 0, i64 0), align 8, !dbg !318 %105 = add i64 %pgocount1, 1, !dbg !318 store i64 %105, i64* getelementptr inbounds ([4 x i64], [4 x i64]*
... @_profc_Z12targetBranchii, i64 0, i64 0), align 8, !dbg !318 %106 = load i32, i32* %i.addr, align 4, !dbg !318
%rem6 = srem i32 %106, 5, !dbg !319
 %mul7 = mul nsw i32 %rem6, 9, !dbg !320
%107 = load i32, i32* %result, align 4, !dbg !321
%add8 = add nsw i32 %107, %mul7, !dbg !321
store i32 %add8, i32* %result, align 4, !dbg !321 %108 = load i32, i32* %i.addr, align 4, !dbg !322
%shr = ashr i32 %108, 3, !dbg !323
%109 = load i32, i32* %result, align 4, !dbg !324
%xor9 = xor i32 %109, %shr, !dbg !324
store i32 %xor9, i32* %result, align 4, !dbg !324
%110 = load i32, i32* %result, align 4, !dbg !325
 %mul10 = mul nsw i32 %110, 2, !dbg !325
store i32 %mul10, i32* %result, align 4, !dbg !325 %111 = load i32, i32* %i.addr, align 4, !dbg !326
 %mul11 = mul nsw i32 %111, 2, !dbg !326
store i32 %mul11, i32* %i.addr, align 4, !dbg !326
%112 = load i32, i32* %i.addr, align 4, !dbg !327
 %rem12 = srem i32 %112, 14, !dbg !328
 %mul13 = mul nsw i32 %rem12, 243, !dbg !329
%113 = load i32, i32* %result, align 4, !dbg !330
%xor14 = xor i32 %113, %mul13, !dbg !330
store i32 %xor14, i32* %result, align 4, !dbg !330
%114 = load i32, i32* %result, align 4, !dbg !331
%div15 = sdiv i32 %114, 3, !dbg !332
store i32 %div15, i32* %i.addr, align 4, !dbg !333 %115 = load i32, i32* %i.addr, align 4, !dbg !334
 %rem16 = srem i32 %115, 5, !dbg !335
%mul17 = mul nsw i32 %rem16, 9, !dbg !336
%116 = load i32, i32* %result, align 4, !dbg !337
%add18 = add nsw i32 %116, %mul17, !dbg !337
store i32 %add18, i32* %result, align 4, !dbg !337
%117 = load i32, i32* %i.addr, align 4, !dbg !338
%shr19 = ashr i32 %117, 1, !dbg !339
```

%118 = load i32, i32* %result, align 4, !dbg !340

store i32 %xor20, i32* %result, align 4, !dbg !340

%xor20 = xor i32 %118, %shr19, !dbg !340

br label %sw.epilog, !dbg !341

```
sw.bb21:
%119 = alloca i32, align 4, !dbg !342
%120 = load i32, i32* %119, align 4, !dbg !342
 %121 = alloca i32, align 4, !dbg !342
%122 = load i32, i32* %121, align 4, !dbg !342
%123 = alloca i32, align 4, !dbg !342
%124 = load i32, i32* %123, align 4, !dbg !342
%125 = alloca i32, align 4, !dbg !342
%126 = load i32, i32* %125, align 4, !dbg !342
%127 = alloca i32, align 4, !dbg !342
%128 = load i32, i32* %127, align 4, !dbg !342
%129 = alloca i32, align 4, !dbg !342
%130 = load i32, i32* %129, align 4, !dbg !342
%131 = alloca i32, align 4, !dbg !342
%132 = load i32, i32* %131, align 4, !dbg !342
%133 = alloca i32, align 4, !dbg !342
%134 = load i32, i32* %133, align 4, !dbg !342
%135 = alloca i32, align 4, !dbg !342
%136 = load i32, i32* %135, align 4, !dbg !342
%137 = alloca i32, align 4, !dbg !342
%138 = load i32, i32* %137, align 4, !dbg !342
%139 = alloca i32, align 4, !dbg !342
%140 = load i32, i32* %139, align 4, !dbg !342
%141 = alloca i32, align 4, !dbg !342
%142 = load i32, i32* %141, align 4, !dbg !342
%143 = alloca i32, align 4, !dbg !342
%144 = load i32, i32* %143, align 4, !dbg !342
%145 = alloca i32, align 4, !dbg !342
%146 = load i32, i32* %145, align 4, !dbg !342
%147 = mul i32 %146, 42, !dbg !342
%148 = mul i32 %147, 42, !dbg !342
%149 = mul i32 %148, 42, !dbg !342
%150 = mul i32 %149, 42, !dbg !342
%151 = mul i32 %150, 42, !dbg !342
%152 = add i32 %151, 42, !dbg !342
%153 = add i32 %152, 42, !dbg !342
%154 = add i32 %153, 42, !dbg !342
%155 = add i32 %154, 42, !dbg !342
%156 = add i32 %155, 42, !dbg !342
%157 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %157, align 4, !dbg !342
%158 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %158, align 4, !dbg !342 %159 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %159, align 4, !dbg !342
%160 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %160, align 4, !dbg !342
%161 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %161, align 4, !dbg !342
%162 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %162, align 4, !dbg !342 %163 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %163, align 4, !dbg !342
%164 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %164, align 4, !dbg !342 %165 = alloca i32, align 4, !dbg !342
store volatile i32 42, i32* %165, align 4, !dbg !342
%pgocount2 = load i64, i64* getelementptr inbounds ([4 x i64], [4 x i64]*
... @_profc_Z12targetBranchii, i64 0, i64 1), align 8, !dbg !342 %166 = add i64 %pgocount2, 1, !dbg !342
store i64 %166, i64* getelementptr inbounds ([4 x i64], [4 x i64]* ... @_profc_Z12targetBranchii, i64 0, i64 1), align 8, !dbg !342 %167 = load i32, i32* %i.addr, align 4, !dbg !342
%rem22 = srem i32 %167, 7, !dbg !343
%mul23 = mul nsw i32 %rem22, 91, !dbg !344
%168 = load i32, i32* %result, align 4, !dbg !345
%add24 = add nsw i32 %168, %mul23, !dbg !345
store i32 %add24, i32* %result, align 4, !dbg !345
%169 = load i32, i32* %i.addr, align 4, !dbg !346
%shr25 = ashr i32 %169, 2, !dbg !347
%170 = load i32, i32* %result, align 4, !dbg !348
%xor26 = xor i32 %170, %shr25, !dbg !348
store i32 %xor26, i32* %result, align 4, !dbg !348 %171 = load i32, i32* %i.addr, align 4, !dbg !349
%rem27 = srem i32 %171, 5, !dbg !350
%idxprom28 = sext i32 %rem27 to i64, !dbg !351
%arrayidx29 = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
 .. %idxprom28, !dbg !351
%172 = load i32, i32* %arrayidx29, align 4, !dbg !351
%173 = load i32, i32* %result, align 4, !dbg !352
%sub = sub nsw i32 %173, %172, !dbg !352
store i32 %sub, i32* %result, align 4, !dbg !352 %174 = load i32, i32* %result, align 4, !dbg !353
%div30 = sdiv i32 %174, 3, !dbg !353
store i32 %div30, i32* %result, align 4, !dbg !353
```

br label %sw.epilog, !dbg !354