```
entry:
 %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 !259,
... metadata !DIExpression()), !dbg !260
 %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 !261, metadata
... !DIExpression()), !dbg !262
 call void @llvm.dbg.declare(metadata [10 x i32]* %A, metadata !263, metadata
... !DIExpression()), !dbg !267
 %0 = bitcast [10 x i32]* %A to i8*, !dbg !267
 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
... !267
 call void @llvm.dbg.declare(metadata i32* %k, metadata !268, metadata
... !DIExpression()), !dbg !269
 %1 = load i32, i32* %branch.addr, align 4, !dbg !270
 %idxprom = sext i32 %1 to i64, !dbg !271
 %arrayidx = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
... %idxprom, !dbg !271
 %2 = load i32, i32* %arrayidx, align 4, !dbg !271
 store i32 %2, i32* %k, align 4, !dbg !269
 call void @llvm.dbg.declare(metadata i32* %result, metadata !272, metadata
... !DIExpression()), !dbg !273
 store i32 0, i32* %result, align 4, !dbg !273
 %3 = load i32, i32* %k, align 4, !dbg !274
 switch i32 %3, label %sw.default [
 i32 0, label %sw.bb
 i32 1, label %sw.bb5
 i32 2, label %sw.bb21
 ], !dbg !275
```

```
sw.default:
 store i32 -1, i32* %result, align 4, !dbg !324
 br label %sw.epilog, !dbg !325
```

```
sw.bb:
 %4 = load i32, i32* %result, align 4, !dbg !276
 %div = sdiv i32 %4, 3, !dbg !276
 store i32 %div, i32* %result, align 4, !dbg !276
 %5 = load i32, i32* %i.addr, align 4, !dbg !278
 %rem = srem i32 %5, 14, !dbg !279
 %mul = mul nsw i32 %rem, 243, !dbg !280
 %6 = load i32, i32* %result, align 4, !dbg !281
 %xor = xor i32 %6, %mul, !dbg !281
 store i32 %xor, i32* %result, align 4, !dbg !281
 %7 = load i32, i32* %i.addr, align 4, !dbg !282
 %rem2 = srem i32 %7, 10, !dbg !283
 %idxprom3 = sext i32 %rem2 to i64, !dbg !284
 %arrayidx4 = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
... %idxprom3, !dbg !284
 %8 = load i32, i32* %arrayidx4, align 4, !dbg !284
 %9 = load i32, i32* %result, align 4, !dbg !285
 %add = add nsw i32 %9, %8, !dbg !285
 store i32 %add, i32* %result, align 4, !dbg !285
 br label %sw.epilog, !dbg !286
```

```
sw.bb5:
 %10 = load i32, i32* %i.addr, align 4, !dbg !287
 %rem6 = srem i32 %10, 5, !dbg !288
 %mul7 = mul nsw i32 %rem6, 9, !dbg !289
 %11 = load i32, i32* %result, align 4, !dbg !290
 %add8 = add nsw i32 %11, %mul7, !dbg !290
 store i32 %add8, i32* %result, align 4, !dbg !290
 %12 = load i32, i32* %i.addr, align 4, !dbg !291
 %shr = ashr i32 %12, 3, !dbg !292
 %13 = load i32, i32* %result, align 4, !dbg !293
 %xor9 = xor i32 %13, %shr, !dbg !293
 store i32 %xor9, i32* %result, align 4, !dbg !293
 %14 = load i32, i32* %result, align 4, !dbg !294
 %mul10 = mul nsw i32 %14, 2, !dbg !294
 store i32 %mul10, i32* %result, align 4, !dbg !294
 %15 = load i32, i32* %i.addr, align 4, !dbg !295
 %mul11 = mul nsw i32 %15, 2, !dbg !295
 store i32 %mul11, i32* %i.addr, align 4, !dbg !295
 %16 = load i32, i32* %i.addr, align 4, !dbg !296
 %rem12 = srem i32 %16, 14, !dbg !297
 %mul13 = mul nsw i32 %rem12, 243, !dbg !298
 %17 = load i32, i32* %result, align 4, !dbg !299
 %xor14 = xor i32 %17, %mul13, !dbg !299
 store i32 %xor14, i32* %result, align 4, !dbg !299
 %18 = load i32, i32* %result, align 4, !dbg !300
 %div15 = sdiv i32 %18, 3, !dbg !301
 store i32 %div15, i32* %i.addr, align 4, !dbg !302
 %19 = load i32, i32* %i.addr, align 4, !dbg !303
 %rem16 = srem i32 %19, 5, !dbg !304
 %mul17 = mul nsw i32 %rem16, 9, !dbg !305
 %20 = load i32, i32* %result, align 4, !dbg !306
 %add18 = add nsw i32 %20, %mul17, !dbg !306
 store i32 %add18, i32* %result, align 4, !dbg !306
 %21 = load i32, i32* %i.addr, align 4, !dbg !307
 %shr19 = ashr i32 %21, 1, !dbg !308
 %22 = load i32, i32* %result, align 4, !dbg !309
 %xor20 = xor i32 %22, %shr19, !dbg !309
 store i32 %xor20, i32* %result, align 4, !dbg !309
 br label %sw.epilog, !dbg !310
```

```
sw.bb21:
 %23 = load i32, i32* %i.addr, align 4, !dbg !311
 %rem22 = srem i32 %23, 7, !dbg !312
 %mul23 = mul nsw i32 %rem22, 91, !dbg !313
 %24 = load i32, i32* %result, align 4, !dbg !314
 %add24 = add nsw i32 %24, %mul23, !dbg !314
 store i32 %add24, i32* %result, align 4, !dbg !314
 %25 = load i32, i32* %i.addr, align 4, !dbg !315
 %shr25 = ashr i32 %25, 2, !dbg !316
 %26 = load i32, i32* %result, align 4, !dbg !317
 %xor26 = xor i32 %26, %shr25, !dbg !317
 store i32 %xor26, i32* %result, align 4, !dbg !317
 %27 = load i32, i32* %i.addr, align 4, !dbg !318
 %rem27 = srem i32 %27, 5, !dbg !319
 %idxprom28 = sext i32 %rem27 to i64, !dbg !320
 %arrayidx29 = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
... %idxprom28, !dbg !320
 %28 = load i32, i32* %arrayidx29, align 4, !dbg !320
 %29 = load i32, i32* %result, align 4, !dbg !321
 %sub = sub nsw i32 %29, %28, !dbg !321
 store i32 %sub, i32* %result, align 4, !dbg !321
 %30 = load i32, i32* %result, align 4, !dbg !322
 %div30 = sdiv i32 %30, 3, !dbg !322
 store i32 %div30, i32* %result, align 4, !dbg !322
 br label %sw.epilog, !dbg !323
```

```
sw.epilog:
 %31 = load i32, i32* %result, align 4, !dbg !326
 ret i32 %31, !dbg !327
```

CFG for '_Z12targetBranchii' function