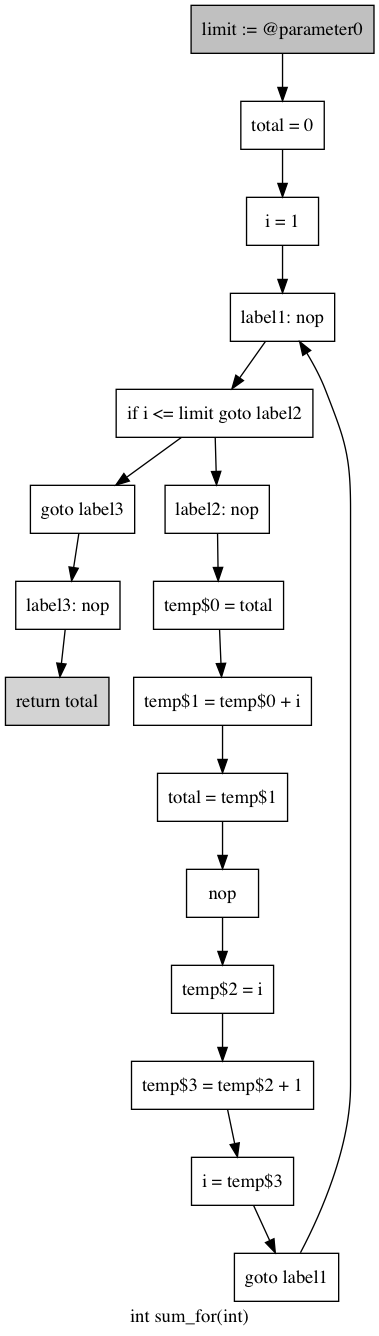
*/\*\*\*  
 \* Sum using for loop  
 \** ***@param limit****: Limit to sum until  
 \** ***@return*** *Sum until limit  
 \*/***public static int** sum\_for(**int** limit){  
 **int** total = 0;  
 **for**(**int** i=1; i<=limit; i++) {  
 total += i;  
 }  
 **return** total;  
}

public static int sum\_for(int)  
{  
 int limit, total, i, temp$0, temp$1, temp$2, temp$3;  
  
 limit := @parameter0: int;  
  
 total = 0;  
  
 i = 1;  
  
 label1:  
 nop;  
  
 if i <= limit goto label2;  
  
 goto label3;  
  
 label2:  
 nop;  
  
 temp$0 = total;  
  
 temp$1 = temp$0 + i;  
  
 total = temp$1;  
  
 nop;  
  
 temp$2 = i;  
  
 temp$3 = temp$2 + 1;  
  
 i = temp$3;  
  
 goto label1;  
  
 label3:  
 nop;  
  
 return total;  
}



E:

X: limit

E: limit

X: total, limit

E: total, limit

X: total, i, limit

E: total, i, limit

X: total, i, limit

E: total, i, limit

X: total, i, limit

E: total, i, limit

X: total, i, limit

E: total

X: total

E: total

X: total

E: limit, i. total

X: limit, temp$0, i

E: total

X:

E: limit, temp$0, i

X: limit, I, temp$1

E: limit, I, temp$1

X: total, limit, i

E: total, limit, i

X: total, limit, i

E: total, limit, i

X: total, limit, temp$2

E: total, limit, temp$2

X: total, limit, temp$3

E: total, limit, temp$3

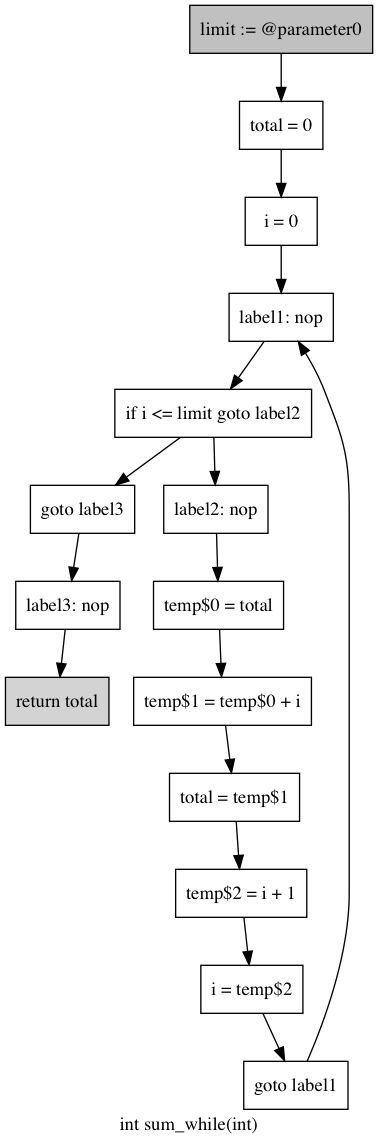
X: total, i, limit

E: total, i, limit

X: total, i, limit

*/\*\*\*  
 \* Sum using while loop  
 \** ***@param limit****: Limit to sum until  
 \** ***@return*** *Sum until limit  
 \*/***public static int** sum\_while(**int** limit) {  
 **int** total = 0, i = 0;  
 **while**(i <= limit){  
 total += i;  
 ++i;  
 }  
 **return** total;  
}

public static int sum\_while(int)  
{  
 int limit, total, i, temp$0, temp$1, temp$2;  
  
 limit := @parameter0: int;  
  
 total = 0;  
  
 i = 0;  
  
 label1:  
 nop;  
  
 if i <= limit goto label2;  
  
 goto label3;  
  
 label2:  
 nop;  
  
 temp$0 = total;  
  
 temp$1 = temp$0 + i;  
  
 total = temp$1;  
  
 temp$2 = i + 1;  
  
 i = temp$2;  
  
 goto label1;  
  
 label3:  
 nop;  
  
 return total;  
}



E:

X: limit

E: limit

X: total, limit

E: total, limit

X: total, i, limit

E: total

X:

E: total

X: total

E: total, i, limit

X: total, i, limit

E: total, i, limit

X: total, i, limit

E: total, i, limit

X: total, i, limit

E: total

X: total

E: limit, i. total

X: limit, temp$0, i

E: limit, temp$0, i

X: limit, i, temp$1

E: limit, I, temp$1

X: total, limit, i

E: total, limit, i

X: total, limit, temp$2

E: total, limit, temp$2

X: total, i, limit

E: total, i, limit

X: total, i, limit

*/\*\*\*  
 \* Sum using mathematical formula  
 \** ***@param limit****: Limit to sum until  
 \** ***@return*** *Sum until limit  
 \*/***public static int** sum\_math(**int** limit) {  
 **return** limit \* (limit + 1) / 2;  
}

E: limit

X: limit, temp$0

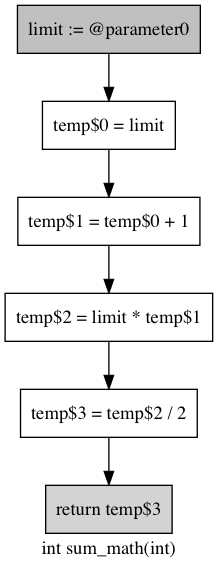
public static int sum\_math(int)  
{  
 int limit, temp$0, temp$1, temp$2, temp$3;  
  
 limit := @parameter0: int;  
  
 temp$0 = limit;  
  
 temp$1 = temp$0 + 1;  
  
 temp$2 = limit \* temp$1;  
  
 temp$3 = temp$2 / 2;  
  
 return temp$3;  
}

E: limit, temp$1

X: limit, temp$2

E: temp$2

X: temp$3



E:

X: limit

E: limit, temp$0

X: limit, temp$1

E: temp$3

X:

**public static int** something\_different(**int** x) {  
 **int** y, z;  
 **while** (x > 2) {  
 y = x/2;  
 **if** (y > 3) x = x - y;  
 z = x - 4;  
 **if** (z > 0) x = x/2;  
 z = z - 1;  
 }  
 **return** x;  
}

E: x

X: x

public static int something\_different(int)  
{  
 int x, y, z, temp$0, temp$1, temp$2, temp$3, temp$4;  
 x := @parameter0: int;  
  
 label1:  
 nop;  
 if x > 2 goto label2;  
 goto label7;  
  
 label2:  
 nop;  
 temp$0 = x / 2;  
 y = temp$0;  
 if y > 3 goto label3;  
 goto label4;  
  
 label3:  
 nop;  
 temp$1 = x - y;  
 x = temp$1;  
  
 label4:  
 nop;  
 temp$2 = x - 4;  
 z = temp$2;  
 if z > 0 goto label5;  
 goto label6;  
  
 label5:  
 nop;  
 temp$3 = x / 2;  
 x = temp$3;  
  
 label6:  
 nop;  
 temp$4 = z - 1;  
 z = temp$4;  
 goto label1;  
  
 label7:  
 nop;  
 return x;  
}

E: x

X:

E: x, z

X: x, z

E: z, temp$3

X: x, z

E: x, z

X: z, temp$3

E: x, z

X: x, z

E: x, z

X: x, z

E: x, z

X: x, z

E: x, temp$2

X: x, z

E: x

X: x, temp$2

E: x

X: x

E: temp$1

X: x

E: x, y

X: temp$1

E: x, y

X: x, y

E: x

X: x

E: x, y

X: x, y

E: x, z

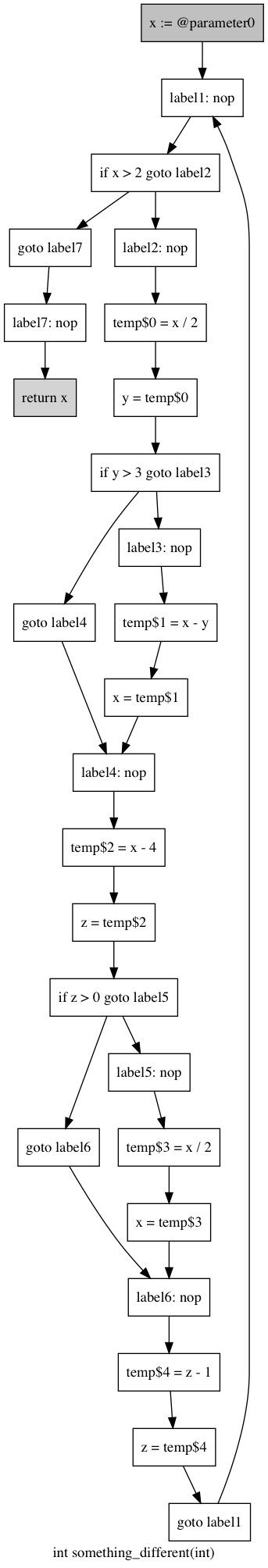
X: x, temp$4

E: x, temp$4

Z: x

E: x

X: x



E:

X: x

E: x, temp$0

X: x, y

XX: x

E: x

X: x, temp$0

E: x

X: x

E: x

X: x

E: x

X: x

E: x

X: x