```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in autumn

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in autumn

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in borland

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in borland

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in bw

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in bw

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in colorful

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in colorful

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in default

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in default

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in emacs

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in emacs

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in friendly

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in friendly

```
add %eax  %ebx
shl %cl  %ebx
jc target
```

Figure 1: GAS assembly in fruity

```
int     0       0
while     0
  if    2    0
```

Figure 2: C source code in fruity

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in manni

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in manni

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in monokai

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in monokai

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in murphy

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in murphy

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in native

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in native

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in pastie

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in pastie

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in perldoc

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in perldoc

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in rrt

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in rrt

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in tango

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in tango

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in trac

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in trac

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in vim

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in vim

```
add %eax, %ebx
shl %cl, %ebx
jc target
```

Figure 1: GAS assembly in vs

```
int c = 0, sum = 0;
while (c > 0) {
  if (c%2 == 0) sum += c;
  c--;
}
assert (sum == input);
```

Figure 2: C source code in vs