CMSC 216
Introduction to Computer Systems
Lecture 13
Assembly Language, cont.

Administrivia

- Continue reading Bryant and O'Hallaron Section 4.1 (Y86 subset) and Chapter 3, for more info on IA-32 instruction set architecture

Chapters 3 and 4.1, Bryant and O'Hallaron

Assembly Language (cont.)

Translating C to Y86

- This process is not always straightforward
- Even simple statements like "**`a = b * c;`**" can require several instructions:

```
irmovl B,%eax
mrmovl 0(%eax),%ebx
irmovl C,%eax
mrmovl 0(%eax),%ecx
multl  %ebx,%ecx
irmovl A,%eax
rmmovl %ecx,0(%eax)
```

Register spilling

- We only have a limited number of registers
- What happens if we need to keep more than 8 values around at once?
- If we run out of registers to store our data, we need to use memory to store values
- We can use the stack or define static arrays (as we'll see later)

Translating loops

- What about C code like this?

```
do {
   ...
} while (condition);
```

- This can be translated simply:

```
Loop: ... # begin loop body
          # evaluate condition
          je Loop # jump back if true
```

- Note that you can use whatever jump instruction/condition is appropriate

Loop Example

- Assembler output:

```
0x000: f208           |           rdint  %eax
0x002: 308700000000   |           irmovl $0,%edi    # consistent zero
0x008: 308600000000   |           irmovl $0,%esi    # sum = 0
0x00e: 6070           |           addl   %edi,%eax
0x010: 7320000000     |           je     EndLoop
                      |
0x015: 6006           | Loop:     addl   %eax,%esi  # sum += n
0x017: f208           |           rdint  %eax
0x019: 6070           |           addl   %edi,%eax
0x01b: 7415000000     |           jne    Loop
                      |
0x020: f368           | EndLoop:  wrint  %esi
0x022: 30830a000000   |           irmovl $10,%ebx
0x028: f138           |           wrch   %ebx
0x02a: 10             |           halt
```

- Simulator output:

```
$ echo 1 4 9 16 25 0 | yis io.yo
55
Stopped in 29 steps at PC = 0x2b.  Exception 'HLT', CC Z=1 S=0 O=0
...
```

Translating while loops

- while loops are a bit different; but these can be written as do-while loops, with a little modification

```
while (condition)
  do_something();
```

```
if (condition) {
  do {
    do_something();
  } while(condition);
}
```

```
if (! condition)
  goto EndWhile;
do {
  do_something();
} while (condition);
EndWhile:
```

Translating while loops, cont.

- Assuming **%eax** always holds the value of our condition (and we keep 0 stored in **%edi**):

```
        irmovl $0,%edi
        addl %edi,%eax
        je EndWhile
Loop:   # code for do_something();
        addl %edi,%eax
        jne Loop
EndWhile:
```

Translating for loops

- For loops are very similar to while loops; we can use this fact to convert a for loop into a form we know how to work with
- How do we convert this into a while loop?

```
for (init; cond; incr)
  body;
```

```
init;
while (cond) {
  body;
  incr;
}
```

Factorial example

- Consider the following program:

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  while (i <= n)
    f *= i++;
  printf("%d\n", f);
  return 0;
}
```

- How would we convert this to a Y86 program?

Translating our factorial example

- Because we only have three variables, we can get away with doing all our work in registers, rather than storing things in memory
- The **`scanf()`** and **`printf()`** calls can be easily replaced by **`rdint`** and **`wrint`**/**`wrch`** instructions
- The main work involved here is just translating the while loop to the do-while format we need to convert to assembly

Translated into goto-code

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  if (i > n)
    goto EndWhile;
 Loop:
  f *= i++;
  if (i <= n)
    goto Loop;
 EndWhile:
  printf("%d\n", f);
  return 0;
}
```

Breaking apart compound statements

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  if (i > n)
    goto EndWhile;
 Loop:
  f *= i++;
  if (i <= n)
    goto Loop;
 EndWhile:
  printf("%d\n", f);
  return 0;
}
```

Breaking apart compound statements

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  if (i > n)
    goto EndWhile;
 Loop:
  f = f * i++;
  if (i <= n)
    goto Loop;
 EndWhile:
  printf("%d\n", f);
  return 0;
}
```

Breaking apart compound statements

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  if (i > n)
    goto EndWhile;
 Loop:
  f = f * i;
  i = i + 1;
  if (i <= n)
    goto Loop;
 EndWhile:
  printf("%d\n", f);
  return 0;
}
```

Beginning "compilation"

- We'll map registers to variables: %eax for i, %ebx for f, and %ecx for n
  - Remember why we can do this here?
- Now, let's translate all the non-branches into the assembly code we know.

Starting compilation

```
#include <stdio.h>

int main() {
  int i, f, n;
  scanf("%d", &n);
  f = 1;
  i = 1;
  if (i > n)
    goto EndWhile;
 Loop:
  f = f * i;
  i = i + 1;
  if (i <= n)
    goto Loop;
 EndWhile:
  printf("%d\n", f);
  return 0;
}
```

```
          rdint   %ecx
          irmovl $1,%ebx
          irmovl $1,%eax
          if (%eax > %ecx)
            goto EndWhile;

Loop:     multl   %eax,%ebx
          irmovl $1,%edi
          addl    %edi,%eax
          if (%eax <= %ecx)
            goto Loop;

EndWhile: wrint   %ebx
          irmovl $10,%edi
          wrch    %edi
          halt
```

Writing branches

- To change the conditional jumps, we'll have to remember a few things:
  - we'll need to operate on copies of our data; this means we'll need to use an extra register
  - we'll need to perform an arithmetic operation before the jump
  - we need to select the correct jump instruction to operate correctly

Starting compilation

```
            rdint   %ecx
            irmovl  $1,%ebx
            irmovl  $1,%eax
            if (%eax > %ecx)
              goto EndWhile;

Loop:       multl   %eax,%ebx
            irmovl  $1,%edi
            addl    %edi,%eax
            if (%eax <= %ecx)
              goto Loop;

EndWhile:   wrint   %ebx
            irmovl  $10,%edi
            wrch    %edi
            halt
```

```
            rdint   %ecx
            irmovl  $1,%ebx
            irmovl  $1,%eax
            rrmovl  %eax,%edi
            subl    %ecx,%edi
            jg      EndWhile

Loop:       multl   %eax,%ebx
            irmovl  $1,%edi
            addl    %edi,%eax
            if (%eax <= %ecx)
              goto Loop;

EndWhile:   wrint   %ebx
            irmovl  $10,%edi
            wrch    %edi
            halt
```

Starting compilation

```
            rdint   %ecx
            irmovl  $1,%ebx
            irmovl  $1,%eax
            if (%eax > %ecx)
              goto EndWhile;

Loop:       multl   %eax,%ebx
            irmovl  $1,%edi
            addl    %edi,%eax
            if (%eax <= %ecx)
              goto Loop;

EndWhile:   wrint   %ebx
            irmovl  $10,%edi
            wrch    %edi
            halt
```

```
            rdint   %ecx
            irmovl  $1,%ebx
            irmovl  $1,%eax
            rrmovl  %eax,%edi
            subl    %ecx,%edi
            jg      EndWhile

Loop:       multl   %eax,%ebx
            irmovl  $1,%edi
            addl    %edi,%eax
            rrmovl  %eax,%edi
            subl    %ecx,%edi
            jle     Loop

EndWhile:   wrint   %ebx
            irmovl  $10,%edi
            wrch    %edi
            halt
```

A working Y86 program

```
            rdint   %ecx          # scanf("%d", &n);
            irmovl  $1,%ebx       # f = 1;
            irmovl  $1,%eax       # i = 1;
            rrmovl  %eax,%edi     # tmp = i;
            subl    %ecx,%edi     # tmp -= n;
            jg      EndWhile      # if (tmp > 0) goto EndWhile;

Loop:       multl   %eax,%ebx     # f *= i;
            irmovl  $1,%edi       # tmp = 1;
            addl    %edi,%eax     # i += tmp;
            rrmovl  %eax,%edi     # tmp = i;
            subl    %ecx,%edi     # tmp -= n;
            jle     Loop          # if (tmp <= 0) goto Loop;

EndWhile:   wrint   %ebx          # printf("%d", f);
            irmovl  $10,%edi      # tmp = '\n';
            wrch    %edi          # printf("%c", tmp);
            halt                  # return 0;
```

Assembly

```
0x000: f218           |             rdint  %ecx
0x002: 308301000000   |             irmovl $1,%ebx
0x008: 308001000000   |             irmovl $1,%eax
0x00e: 2007           |             rrmovl %eax,%edi
0x010: 6117           |             subl   %ecx,%edi
0x012: 762a000000     |             jg     EndWhile
                      |
0x017: 6403           | Loop:       multl  %eax,%ebx
0x019: 308701000000   |             irmovl $1,%edi
0x01f: 6070           |             addl   %edi,%eax
0x021: 2007           |             rrmovl %eax,%edi
0x023: 6117           |             subl   %ecx,%edi
0x025: 7117000000     |             jle    Loop
                      |
0x02a: f338           | EndWhile:   wrint  %ebx
0x02c: 30870a000000   |             irmovl $10,%edi
0x032: f178           |             wrch   %edi
0x034: 10             |             halt
```

The program in memory

```
0x00: f2 18 30 83 01 00 00 00
0x08: 30 80 01 00 00 00 20 07
0x10: 61 17 76 2a 00 00 00 64
0x18: 03 30 87 01 00 00 00 60
0x20: 70 20 07 61 17 71 17 00
0x28: 00 00 f3 38 30 87 0a 00
0x30: 00 00 f1 78 10 00 00 00
```

Assembly

```
0x000: f218           |             rdint  %ecx
0x002: 308301000000   |             irmovl $1,%ebx
0x008: 308001000000   |             irmovl $1,%eax
0x00e: 2007           |             rrmovl %eax,%edi
0x010: 6117           |             subl   %ecx,%edi
0x012: 762a000000     |             jg     EndWhile
                      |
0x017: 6403           | Loop:       multl  %eax,%ebx
0x019: 308701000000   |             irmovl $1,%edi
0x01f: 6070           |             addl   %edi,%eax
0x021: 2007           |             rrmovl %eax,%edi
0x023: 6117           |             subl   %ecx,%edi
0x025: 7117000000     |             jle    Loop
                      |
0x02a: f338           | EndWhile:   wrint  %ebx
0x02c: 30870a000000   |             irmovl $10,%edi
0x032: f178           |             wrch   %edi
0x034: 10             |             halt
```

Assembly

```
0x000: |            rdint  %ecx
0x002: |            irmovl $1,%ebx
0x008: |            irmovl $1,%eax
0x00e: |            rrmovl %eax,%edi
0x010: |            subl   %ecx,%edi
0x012: |            jg     EndWhile
       |
0x017: | Loop:      multl  %eax,%ebx
0x019: |            irmovl $1,%edi
0x01f: |            addl   %edi,%eax
0x021: |            rrmovl %eax,%edi
0x023: |            subl   %ecx,%edi
0x025: |            jle    Loop
       |
0x02a: | EndWhile:  wrint  %ebx
0x02c: |            irmovl $10,%edi
0x032: |            wrch   %edi
0x034: |            halt
```

Label Translation

```
0x000: |            rdint  %ecx
0x002: |            irmovl $1,%ebx
0x008: |            irmovl $1,%eax
0x00e: |            rrmovl %eax,%edi
0x010: |            subl   %ecx,%edi
0x012: |            jg     0x2a
       |
0x017: | Loop:      multl  %eax,%ebx
0x019: |            irmovl $1,%edi
0x01f: |            addl   %edi,%eax
0x021: |            rrmovl %eax,%edi
0x023: |            subl   %ecx,%edi
0x025: |            jle    0x17
       |
0x02a: | EndWhile:  wrint  %ebx
0x02c: |            irmovl $10,%edi
0x032: |            wrch   %edi
0x034: |            halt
```

Final Factorial Program

```
0x000: |            rdint  %ecx
0x002: |            irmovl $1,%ebx
0x008: |            irmovl $1,%eax
0x00e: |            rrmovl %eax,%edi
0x010: |            subl   %ecx,%edi
0x012: |            jg     0x2a
       |
0x017: | Loop:      multl  %eax,%ebx
0x019: |            irmovl $1,%edi
0x01f: |            addl   %edi,%eax
0x021: |            rrmovl %eax,%edi
0x023: |            subl   %ecx,%edi
0x025: |            jle    0x17
       |
0x02a: | EndWhile:  wrint  %ebx
0x02c: |            irmovl $10,%edi
0x032: |            wrch   %edi
0x034: |            halt
```

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	0
%ecx	0
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x000
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000: | rdint   %ecx
0x002: | irmovl $1,%ebx
0x008: | irmovl $1,%eax
0x00e: | rrmovl %eax,%edi
0x010: | subl   %ecx,%edi
0x012: | jg     0x2a
0x017: | multl  %eax,%ebx
0x019: | irmovl $1,%edi
0x01f: | addl   %edi,%eax
0x021: | rrmovl %eax,%edi
0x023: | subl   %ecx,%edi
0x025: | jle    0x17
0x02a: | wrint  %ebx
0x02c: | irmovl $10,%edi
0x032: | wrch   %edi
0x034: | halt
```

Register name	Value
%eax	0
%ecx	0
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

Input is: 4

PC	0x002
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

Input is: 4

PC	0x002
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	0
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x008
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x008
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	0
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x00e
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x00e
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x010
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x010
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x012
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x012
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

```
jg:
~(SF ^ OF) & ~ZF = 0
```

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-3

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	1
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	2

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	2

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x025
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x02a
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

```
jle:
(SF ^ OF)  |  ZF = 1
```

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	1
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-2

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	2
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	3

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	3

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x025
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x02a
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

```
jle:
(SF ^ OF)  |  ZF = 1
```

PC	0x017
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	2
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	-1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	3
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	4

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	4

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x025
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x02a
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

```
jle:
(SF ^ OF)  |  ZF = 1
```

PC	0x017
CC: ZF	1
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	6
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x019
CC: ZF	0
CC: SF	1
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x019
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	0

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x01f
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	4
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x021
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	5

PC	0x023
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	5

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x025
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x02a
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

```
jle:
(SF ^ OF)  |  ZF = 0
```

PC	0x02a
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

PC	0x02c
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

Output is: **24**

PC	0x02c
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	1

Output is: 24

PC	0x032
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

Output is: 24

PC	0x032
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

Output is: **24**

PC	0x034
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

Output is: 24

PC	0x034
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

```
0x000:  |  rdint   %ecx
0x002:  |  irmovl  $1,%ebx
0x008:  |  irmovl  $1,%eax
0x00e:  |  rrmovl  %eax,%edi
0x010:  |  subl    %ecx,%edi
0x012:  |  jg      0x2a
0x017:  |  multl   %eax,%ebx
0x019:  |  irmovl  $1,%edi
0x01f:  |  addl    %edi,%eax
0x021:  |  rrmovl  %eax,%edi
0x023:  |  subl    %ecx,%edi
0x025:  |  jle     0x17
0x02a:  |  wrint   %ebx
0x02c:  |  irmovl  $10,%edi
0x032:  |  wrch    %edi
0x034:  |  halt
```

Register name	Value
%eax	5
%ecx	4
%edx	0
%ebx	24
%esp	0
%ebp	0
%esi	0
%edi	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

- These are the final states of the PC, CC, and all the registers.
- Does this agree with what **`yis`** tells us?
- Remember, numbers are printed in hex, so $10_{10} = a_{16}$, $24_{10} = 18_{16}$

Register name	Value
`%eax`	5
`%ecx`	4
`%edx`	0
`%ebx`	24
`%esp`	0
`%ebp`	0
`%esi`	0
`%edi`	10

PC	0x035
CC: ZF	0
CC: SF	0
CC: OF	0

Execution

- Yes:

```
$ echo 4 | yis factorial.yo
24
Stopped in 34 steps at PC = 0x35.  Exception 'HLT', CC Z=0 S=0 O=0
Changes to registers:
%eax:   0x00000000      0x00000005
%ecx:   0x00000000      0x00000004
%ebx:   0x00000000      0x00000018
%edi:   0x00000000      0x0000000a

Changes to memory:
```

Some other notes

- Exceptions
  - If your code exits due to a **`halt`** instruction, the exception will be HLT (which is fine)
  - You can also stop your code prematurely by passing a second parameter (max steps) to **`yis`**; this will usually result in the exception AOK
  - But if you manage to move to a bad address (forget to halt?), divide by zero, or just make the simulator otherwise unhappy, you'll get a different exception (e.g., ADR or INS)

Program efficiency

- Our assembly program was not as efficient as it could be
  - we could have optimized away the need for the variable **`i`** by simply decrementing **`n`** until **`n`** was 0
  - we also could have been smarter with our registers, and not overwritten the constant value we used repeatedly (1)
- These things need to be taken into consideration by compilers, and by you in the assembly code you're writing for the project...