ECS 40 **Program #5 (50 pts, 2 hours)** **Winter 2016**
Due: Wednesday, February 17th, 11:59 PM. Executable Name: CPU.out

Filenames (case sensitive): authors.csv, data.cpp, data.h, decoder.cpp, decoder.h, instruction.cpp, instruction.h, instruction2.cpp, instruction2.h, labels.cpp, labels.h, main.cpp, memory.cpp, memory.h, registers.cpp, registers.h, word.cpp, word.h, and Makefile.

fthat involves multiplication, and recursion. More importantly, you will also be implementing many overloaded operators, a linked list, and a memory system that can hold both Instructions and data. As usual, all implementation code should be in the .cpp files. You are free (and encouraged) to use my source code for p4 as your starting point; it is in ~ssdavis/40/p4/SeansSrc on Thursday. In any case, you may wish to refer to my code if you do not understand some of the references in the instructions. There are three new assembly files, test6.s, fac.s, and fib.s. They, and their C files, are attached.

When describing CPUs, a "word" is the size of a data register. Older Intel CPUs had 32-bit (4-bytes) words, and current ones have 64-bit (8 bytes) words. While our program stores instructions as strings, an executable program uses numerical encodings, called "machine code", to indicate opcode and address information. Any byte in RAM can hold data or machine code at any given time. It is up to each program to determine whether to use a byte as part of an instruction or part of some data. Since our CPU simulator is processing assembly code rather than machine code, the access to memory cannot be simulated as simply an array of bytes because our Instruction strings do mimic the memory footprint of their corresponding machine code. When we have read the .s files we have treated each instruction as having used four bytes (hence address += 4) even though the actual strings take much more space. To accommodate the simulated addresses, we have had to accompany each Instruction string with its simulated address.

In the previous CPU programs, the Reader class' array of Instructions, and the memory array, were overlapping views of the same object, RAM memory. This program is going to combine the memory array of ints and the Reader class Instruction array into a single linked list of Word*. The Word class will only contain an int named address, a constructor, a virtual destructor, and an overloaded < operator. The Word class will have the Instruction class, and the Data class publicly derived from it. The Instruction class is the same one as in the previous program except that it no longer has an explicit address. The Data class simply contains an int.

The Memory class will be a singly linked list sorted by address. The list nodes will be class ListNode objects, and have Memory as a friend. The ListNode class is defined in Memory.h, and contains no public methods! Each ListNode will contain a Word* named word, and a ListNode* named next. The two overloaded operator[] of the Memory class return Word* that will be dynamic_cast by the receiving function into either a Instruction* or a Data*. The Memory class will rely on the operator< of the Word class for both sorting and testing for equality.

The following approach to development omits aspects that you should already know, such as placing the proper #includes at the top of files that need them. Rather than depend on the Makefile all of the time, you may wish to use g++ -c with a specific .cpp files to concentrate on getting it to compile. Your code should compile without warnings using the Makefile, and run properly with test5.s after each major step.

1. Create a Word class with only an int named address, and only the methods described below.
   1.1. The address should be private, not protected.
   1.2. The constructor will take an address as its only parameter.
   1.3. The virtual destructor will be empty, but is necessary for any class that will serve as a base class.
   1.4. The overloaded operator< will take a Word as its parameter, and will compare the addresses.
   1.5. For debugging purposes I added a public getAddress(), but you must remove it before submitting your code.
2. Create a Data class that is publicly derived from the Word class, and only the methods described below.
   2.1. The only data member of the class is an int named num.
   2.2. The constructor will only take an int address, and call the Word constructor in its initialization list.
   2.3. There are two get() methods. Both return a reference to num, but one of the methods is const.
   2.4. The overloaded assignment operator should take an int as its parameter.
3. Create the Memory and ListNode classes with only the methods described below.
   3.1. You have to have a forward declaration of the Memory class above the ListNode class so that you can make the Memory class a friend of the ListNode class.
   3.2. The ListNode constructor will take a Word* and a ListNode*.
   3.3. The ListNode destructor must delete the Word*.
   3.4. The Memory constructor will simply set head to NULL.
   3.5. The Memory destructor will delete all of the list's ListNodes.

- 3.6. One of the Memory operator[] will be const, and the other operator[] will not be const.
  - 3.6.1. The methods take an address as their parameter. Since the operator< of Word requires a Word as its parameter, you will need to construct a Word from the address when using that operator.
  - 3.6.2. The methods return a Word&.
  - 3.6.3. Hints: If an address is neither less than another int, nor greater than another int, then it must be the same value. Since the code is identical in both methods, write one first and get it to compile, and the copy and paste its code into the second one.
  - 3.6.4. The only time a specified address will not be found is when a new index into the old memory array is used to store an integer, and thus would be calling the non-const version of the operator. The const version should print and error message "Seg fault at address: <the address>" and exit(1) if an address is not found. You should do your own research to find out which header file provides the prototype for exit(). The non-const version will have to do three things to deal with a missing address:
    - 3.6.4.1. Create a new Data with the specified address.
    - 3.6.4.2. Insert the new Data in the Memory linked list using insert(). Don't use "this" to call insert()!!!
    - 3.6.4.3. Return the new Data as a reference.
- 3.7. The Memory::insert() will take a Word* as its sole parameter. It is assumed that the parameter has been dynamically allocated as either a Data* or an Instruction* before being passed into insert().

4. Replace the memory array with a Memory object. This involves a lot of changes, and it is best to compile often.
- 4.1. Changes in main()
  - 4.1.1. Replace the memory array declaration with a Memory object declaration.
  - 4.1.2. Replace the initialization of memory[1000], with three lines:
    - 4.1.2.1. Declare a Data pointer and set it equal to a Data that has been dynamically allocate with address 1000.
    - 4.1.2.2. Use the overloaded assignment operator to set the data's num to 0.
    - 4.1.2.3. insert() the data into the memory object.
  - 4.1.3. As in 4.1.2, initialize memory[992] to 0.
- 4.2. Changes in decoder.cpp and decoder.h
  - 4.2.1. Change the parameter type of memory for the Decoder class from an int array to a Memory &. A simple Replace All in both files makes this quick and easy. In vi this would be ":%s/int memory[1001]/Memory &memory/g". Passing a non-const reference violates the suggest style, but it makes the use of the operator[] easier. If memory was passed as a pointer, you would have to use (*memory)[address], but as a reference you can use memory[address].
  - 4.2.2. Change the decoder.cpp accesses to the int memory to accesses to Data objects in the Memory object. To access the Data objects contained in Memory will require use of dynamic_casting to a Data reference, e.g. Data &data = dynamic_cast <Data&> (memory[1000]); Note: I forgot to do this in pushl(), and it cost me two hours of debugging!!!
- 4.3. Changes in registers.cpp and registers.h
  - 4.3.1. Make changes similar to the decoder files.
  - 4.3.2. To return the address of the num within a Data object, you can apply the '&' operator on the result of a get(). I wrote this as one long return statement involving the cast and many parentheses, but you may wish to assign the data, and then call get() in a separate statement.
- 4.4. Believe it or not, your program should now compile without errors and run perfectly! Congratulations.

5. Make a duplicate Instruction class for Labels to use.
- 5.1. Make a duplicate of the Instruction class called Instruction2.
  - 5.1.1. Copy instruction.cpp to instruction2.cpp, and instruction.h to instruction2.h
  - 5.1.2. Change the name of the class in the instruction2 files to Instruction2. Another Replace All can do this in seconds.
  - 5.1.3. Remember to change the guard statements at the top of instruction2.h to INSTRUCTION2_H.
  - 5.1.4. Remember to change the #include to instruction2.h at the top of instruction2.cpp.
- 5.2. Changes in labels.cpp and labels.h
  - 5.2.1. Change the array from Instruction to Instruction2 in labels.h.
  - 5.2.2. #include instruction2.h instead of instruction.h.
- 5.3. Changes in the Makefile
  - 5.3.1. Make labels.o dependent on instruction2.h
  - 5.3.2. Add lines to create instruction2.o, and add to the CPU.out dependency and linking lines.

6. Make Instruction class a derived class of Word stored in the Memory object.

- 6.1. Changes in instruction.cpp and instruction.h
  - 6.1.1. Make the class publicly derived from the Word class, and remove the address from the Instruction class.
  - 6.1.2. Change the constructor to take an address as its sole parameter.
  - 6.1.3. Change setInfo() to an overloaded assignment operator that takes a const char* as its parameter.
  - 6.1.4. Eliminate getAddress(), and setAddress().
  - 6.1.5. Add an overloaded operator<< for ostream that takes an Instruction reference as its second parameter and prints the info of the Instruction.
- 6.2. Changes in main()
  - 6.2.1. Eliminate the declaration of the reader object, and substitute the memory object wherever the reader object was used.
  - 6.2.2. Change the statement in main() that prints the Instruction information to use the operator<< of Instruction.
  - 6.2.3. To avoid the seg faults occurring because of the an Instruction::info being deleted twice, eliminate the declaration of the Instruction at the top of main(). Instead, declare a const Instruction& that is set by the return value of fetch(). fetch() no longer has an Instruction* parameter.
- 6.3. Changes in decoder.cpp and decoder.h
  - 6.3.1. execute() and parse() take const Instruction& as a parameter instead of const Instruction*.
- 6.4. Changes in memory.cpp and memory.h
  - 6.4.1. Move the fetch() method from Reader to the Memory class.
    - 6.4.1.1. Because memory.h is #included in registers.h you cannot #include "register.h" in memory.h. Instead, provide a forward declaration for the Register class. You can #include "register.h" in memory.cpp.
    - 6.4.1.2. fetch() no longer takes an Instruction* as a parameter, and returns a const Instruction&.
    - 6.4.1.3. This function can be simplified by calling the operator[] of Memory using (*this)[…] and dynamic casting the result to an Instruction&
  - 6.4.2. Move the operator>> methods from Reader to the Memory class.
    - 6.4.2.1. The insertion into the old Instruction array will be replaced with a call to the Instruction constructor, a call to the operator= of Instruction, and a call to the Memory::insert() method.

7. Making fac.s and fib.s work
- 7.1. The function labels do not begin with an underscore, but they do end with a colon, as do all labels. Only labels end with colons. Alter the methods of Labels, Memory, and Registers to handle this change. There are two new registers used, ebx, and ecx, which are like eax. You will just need to make a few changes to Registers to handle these new registers. I hope you will appreciate how easy this change is. In C, you might have to scour your code to handle such an addition.
- 7.2. Decoder needs to process two new opcodes.
  - 7.2.1. **decl** *operand*: decrement the *operand*. This affects both flags.
  - 7.2.2. **imull** *operand1*, *operand2*: multiplies *operand1* by *operand2*, and puts the result in *operand2*. This affects both flags.
- 7.3. Since leal now can operate on a register other than ebp, Decoder::leal() will have to be modified to call Registers:stringToRegNum(). To make this work, stringToRegNum() will now have to be a public method of Registers.

Further specifications:
- 1.1. const must be used wherever possible in function headings. This includes parameters, return types, and functions themselves. Note that there is no need to label parameters and return types that are passed by value as const.
- 1.2. You may assume that all input will be valid, and not require any form of range checking.
- 1.3. You must use g++ with the –g –Wall –ansi options for compiling and linking. You will lose one point for each warning.
- 1.4. You will find fib.s, fac.s, and my own executable in ~ssdavis/40/p5.

```
[ssdavis@lect1 p5]$ cat test7.c

int fac(int n)
{
  if(n < 1)
    return 1;

  return n * fac(n - 1);
} // fac()


int main()
{
  int a, b, c = 4;

  a = fac(c);
  b = fac(3);
  return a + b;
} // main()
[ssdavis@lect1 p5]$
```

```
[ssdavis@lect1 p5]$ cat test7.s
        .file   "test7.c"
        .text
        .globl  fac
        .type   fac, @function
fac:
.LFB0:
        .cfi_startproc
        pushl   %ebp
        .cfi_def_cfa_offset 8
        .cfi_offset 5, -8
        movl    %esp, %ebp
        .cfi_def_cfa_register 5
        subl    $8, %esp
        cmpl    $0, 8(%ebp)
        jg      .L2
        movl    $1, %eax
        jmp     .L3
.L2:
        movl    8(%ebp), %eax
        subl    $1, %eax
        subl    $12, %esp
        pushl   %eax
        call    fac
        addl    $16, %esp
        imull   8(%ebp), %eax
.L3:
        leave
        .cfi_restore 5
        .cfi_def_cfa 4, 4
        ret
        .cfi_endproc
.LFE0:
        .size   fac, .-fac
        .globl  main
        .type   main, @function
main:
.LFB1:
        .cfi_startproc
        leal    4(%esp), %ecx
        .cfi_def_cfa 1, 0
        andl    $-16, %esp
        pushl   -4(%ecx)
        pushl   %ebp
        .cfi_escape 0x10,0x5,0x2,0x75,0
        movl    %esp, %ebp
        pushl   %ecx
        .cfi_escape 0xf,0x3,0x75,0x7c,0x6
        subl    $20, %esp
        movl    $4, -12(%ebp)
        subl    $12, %esp
        pushl   -12(%ebp)
        call    fac
        addl    $16, %esp
        movl    %eax, -16(%ebp)
        subl    $12, %esp
        pushl   $3
        call    fac
        addl    $16, %esp
        movl    %eax, -20(%ebp)
        movl    -16(%ebp), %edx
        movl    -20(%ebp), %eax
        addl    %edx, %eax
        movl    -4(%ebp), %ecx
        .cfi_def_cfa 1, 0
        leave
        .cfi_restore 5
        leal    -4(%ecx), %esp
        .cfi_def_cfa 4, 4
        ret
        .cfi_endproc
.LFE1:
        .size   main, .-main
        .ident  "GCC: (GNU) 5.3.1 20151207 (Red Hat 5.3.1-2)"
        .section        .note.GNU-stack,"",@progbits
[ssdavis@lect1 p5]$
```

```
[ssdavis@lect1 p5]$ cat test8.c
int fib(int n)
{
  if(n < 2)
    return 1;

  return fib(n - 1) + fib(n - 2);
}  // fib()

int main()
{
  int a, b, c;

  a = 4;
  b = fib(a);
  c = fib(5);
  return b + c + a;
} // main()
[ssdavis@lect1 p5]$


[ssdavis@lect1 p5]$ cat test8.s
        .file   "test8.c"
        .text
        .globl  fib
        .type   fib, @function
fib:
.LFB0:
        .cfi_startproc
        pushl   %ebp
        .cfi_def_cfa_offset 8
        .cfi_offset 5, -8
        movl    %esp, %ebp
        .cfi_def_cfa_register 5
        pushl   %ebx
        subl    $4, %esp
        .cfi_offset 3, -12
        cmpl    $1, 8(%ebp)
        jg      .L2
        movl    $1, %eax
        jmp     .L3
.L2:
        movl    8(%ebp), %eax
        subl    $1, %eax
        subl    $12, %esp
        pushl   %eax
        call    fib
        addl    $16, %esp
        movl    %eax, %ebx
        movl    8(%ebp), %eax
        subl    $2, %eax
        subl    $12, %esp
        pushl   %eax
        call    fib
        addl    $16, %esp
        addl    %ebx, %eax
.L3:
        movl    -4(%ebp), %ebx
        leave
        .cfi_restore 5
        .cfi_restore 3
        .cfi_def_cfa 4, 4
        ret
        .cfi_endproc
.LFE0:
        .size   fib, .-fib
        .globl  main
        .type   main, @function
main:
.LFB1:
        .cfi_startproc
        leal    4(%esp), %ecx
        .cfi_def_cfa 1, 0
        andl    $-16, %esp
        pushl   -4(%ecx)
        pushl   %ebp
        .cfi_escape 0x10,0x5,0x2,0x75,0
        movl    %esp, %ebp
        pushl   %ecx
        .cfi_escape 0xf,0x3,0x75,0x7c,0x6
        subl    $20, %esp
        movl    $4, -12(%ebp)
        subl    $12, %esp
        pushl   -12(%ebp)
        call    fib
        addl    $16, %esp
        movl    %eax, -16(%ebp)
        subl    $12, %esp
        pushl   $5
        call    fib
        addl    $16, %esp
        movl    %eax, -20(%ebp)
        movl    -16(%ebp), %edx
        movl    -20(%ebp), %eax
        addl    %eax, %edx
        movl    -12(%ebp), %eax
        addl    %edx, %eax
        movl    -4(%ebp), %ecx
        .cfi_def_cfa 1, 0
        leave
        .cfi_restore 5
        leal    -4(%ecx), %esp
        .cfi_def_cfa 4, 4
        ret
        .cfi_endproc
.LFE1:
        .size   main, .-main
        .ident  "GCC: (GNU) 5.3.1 20151207 (Red Hat 5.3.1-2)"
        .section        .note.GNU-stack,"",@progbits
[ssdavis@lect1 p5]$
```

```
[ssdavis@lect1 p5]$ CPU.out test7.s
leal 4(%esp), %ecx   eip: 168 eax:   0 ebp: 996 esp: 1000 edx:   0 ebx:   0 ecx: 1004 flags: 192
andl $-16, %esp      eip: 172 eax:   0 ebp: 996 esp: 992 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl -4(%ecx)       eip: 176 eax:   0 ebp: 996 esp: 988 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 180 eax:   0 ebp: 996 esp: 984 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 184 eax:   0 ebp: 984 esp: 984 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ecx           eip: 188 eax:   0 ebp: 984 esp: 980 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $20, %esp       eip: 192 eax:   0 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl $4, -12(%ebp)   eip: 196 eax:   0 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 200 eax:   0 ebp: 984 esp: 948 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl -12(%ebp)      eip: 204 eax:   0 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   0 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 984 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 936 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   4 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 140 eax:   3 ebp: 936 esp: 916 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   3 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   3 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   3 ebp: 936 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   3 ebp: 904 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 140 eax:   2 ebp: 904 esp: 884 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 904 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 872 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 140 eax:   1 ebp: 872 esp: 852 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 872 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 840 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
subl $12, %esp       eip: 140 eax:   0 ebp: 840 esp: 820 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   0 ebp: 840 esp: 816 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   0 ebp: 840 esp: 812 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 840 esp: 808 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 808 esp: 808 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
jg .L2               eip: 120 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl $1, %eax        eip: 124 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
jmp .L3              eip: 156 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
leave                eip: 160 eax:   1 ebp: 840 esp: 812 edx:   0 ebx:   0 ecx: 1004 flags:  64
ret                  eip: 148 eax:   1 ebp: 840 esp: 816 edx:   0 ebx:   0 ecx: 1004 flags:  64
addl $16, %esp       eip: 152 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 148 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 152 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 148 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 152 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   6 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   6 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 148 eax:   6 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 152 eax:   6 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:  24 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:  24 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 208 eax:  24 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 212 eax:  24 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, -16(%ebp) eip: 216 eax:  24 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
```

```
subl $12, %esp       eip: 220 eax:  24 ebp: 984 esp: 948 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl $3             eip: 224 eax:  24 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:  24 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:  24 ebp: 984 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:  24 ebp: 936 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:  24 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:  24 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:  24 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   2 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 140 eax:   2 ebp: 936 esp: 916 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   2 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   2 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 936 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 904 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 140 eax:   1 ebp: 904 esp: 884 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 904 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 872 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 128 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 132 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 136 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:  64
subl $12, %esp       eip: 140 eax:   0 ebp: 872 esp: 852 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 144 eax:   0 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fac             eip: 100 eax:   0 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 872 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 840 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $8, %esp        eip: 112 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $0, 8(%ebp)     eip: 116 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
jg .L2               eip: 120 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl $1, %eax        eip: 124 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
jmp .L3              eip: 156 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
leave                eip: 160 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:  64
ret                  eip: 148 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:  64
addl $16, %esp       eip: 152 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 148 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 152 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   2 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 148 eax:   2 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 152 eax:   2 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
imull 8(%ebp), %eax  eip: 156 eax:   6 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 160 eax:   6 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 228 eax:   6 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 232 eax:   6 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, -20(%ebp) eip: 236 eax:   6 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl -16(%ebp), %edx eip: 240 eax:   6 ebp: 984 esp: 960 edx:  24 ebx:   0 ecx: 1004 flags:   0
movl -20(%ebp), %eax eip: 244 eax:   6 ebp: 984 esp: 960 edx:  24 ebx:   0 ecx: 1004 flags:   0
addl %edx, %eax      eip: 248 eax:  30 ebp: 984 esp: 960 edx:  24 ebx:   0 ecx: 1004 flags:   0
movl -4(%ebp), %ecx  eip: 252 eax:  30 ebp: 984 esp: 960 edx:  24 ebx:   0 ecx: 1004 flags:   0
leave                eip: 256 eax:  30 ebp: 996 esp: 988 edx:  24 ebx:   0 ecx: 1004 flags:   0
leal -4(%ecx), %esp  eip: 260 eax:  30 ebp: 996 esp: 1000 edx:  24 ebx:   0 ecx: 1004 flags:   0
ret                  eip:   0 eax:  30 ebp: 996 esp: 1004 edx:  24 ebx:   0 ecx: 1004 flags:   0
[ssdavis@lect1 p5]$
```

```
[ssdavis@lect1 p5]$ CPU.out test8.s
leal 4(%esp), %ecx   eip: 204 eax:   0 ebp: 996 esp: 1000 edx:   0 ebx:   0 ecx: 1004 flags: 192
andl $-16, %esp      eip: 208 eax:   0 ebp: 996 esp: 992 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl -4(%ecx)       eip: 212 eax:   0 ebp: 996 esp: 988 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 216 eax:   0 ebp: 996 esp: 984 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 220 eax:   0 ebp: 984 esp: 984 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ecx           eip: 224 eax:   0 ebp: 984 esp: 980 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $20, %esp       eip: 228 eax:   0 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl $4, -12(%ebp)   eip: 232 eax:   0 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 236 eax:   0 ebp: 984 esp: 948 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl -12(%ebp)      eip: 240 eax:   0 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib             eip: 100 eax:   0 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 984 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 936 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   0 ebp: 936 esp: 932 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   0 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   4 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   3 ebp: 936 esp: 916 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   3 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib             eip: 100 eax:   3 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   3 ebp: 936 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   3 ebp: 904 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   3 ebp: 904 esp: 900 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   2 ebp: 904 esp: 884 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib             eip: 100 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 904 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 872 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   2 ebp: 872 esp: 868 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   1 ebp: 872 esp: 852 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 872 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 840 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 840 esp: 836 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:  64
ret                  eip: 152 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:  64
addl $16, %esp       eip: 156 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:  64
subl $12, %esp       eip: 172 eax:   0 ebp: 872 esp: 852 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   0 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags:   0
call fib             eip: 100 eax:   0 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 872 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 840 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   0 ebp: 840 esp: 836 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jg .L2               eip: 124 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags: 128
ret                  eip: 180 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags: 128
addl $16, %esp       eip: 184 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 152 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
```

```
addl $16, %esp       eip: 156 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   1 ebp: 904 esp: 884 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   2 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 904 esp: 872 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 872 esp: 872 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 872 esp: 868 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   2 ecx: 1004 flags:  64
ret                  eip: 180 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   2 ecx: 1004 flags:  64
addl $16, %esp       eip: 184 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   3 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 152 eax:   3 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 156 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   4 ebp: 936 esp: 928 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   2 ebp: 936 esp: 928 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   2 ebp: 936 esp: 916 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   2 ebp: 936 esp: 912 edx:   0 ebx:   3 ecx: 1004 flags:   0
call fib             eip: 100 eax:   2 ebp: 936 esp: 908 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 936 esp: 904 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 904 esp: 904 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   2 ebp: 904 esp: 900 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   1 ebp: 904 esp: 884 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   3 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 904 esp: 872 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 872 esp: 872 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 872 esp: 868 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   3 ecx: 1004 flags:  64
ret                  eip: 152 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   3 ecx: 1004 flags:  64
addl $16, %esp       eip: 156 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   0 ebp: 904 esp: 896 edx:   0 ebx:   1 ecx: 1004 flags:  64
subl $12, %esp       eip: 172 eax:   0 ebp: 904 esp: 884 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   0 ebp: 904 esp: 880 edx:   0 ebx:   1 ecx: 1004 flags:   0
call fib             eip: 100 eax:   0 ebp: 904 esp: 876 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 904 esp: 872 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 872 esp: 872 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   0 ebp: 872 esp: 868 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags: 128
jg .L2               eip: 124 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl $1, %eax        eip: 128 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags: 128
jmp .L3              eip: 188 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags: 128
leave                eip: 196 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   1 ecx: 1004 flags: 128
ret                  eip: 180 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   1 ecx: 1004 flags: 128
addl $16, %esp       eip: 184 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   1 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
leave                eip: 196 eax:   2 ebp: 936 esp: 908 edx:   0 ebx:   3 ecx: 1004 flags:   0
ret                  eip: 180 eax:   2 ebp: 936 esp: 912 edx:   0 ebx:   3 ecx: 1004 flags:   0
addl $16, %esp       eip: 184 eax:   2 ebp: 936 esp: 928 edx:   0 ebx:   3 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   5 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
```

```
ret                 eip: 244 eax:   5 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp      eip: 248 eax:   5 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, -16(%ebp) eip: 252 eax:   5 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp      eip: 256 eax:   5 ebp: 984 esp: 948 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl $5            eip: 260 eax:   5 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib            eip: 100 eax:   5 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   5 ebp: 984 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   5 ebp: 936 esp: 936 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   5 ebp: 936 esp: 932 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2              eip: 132 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax  eip: 136 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax       eip: 140 eax:   4 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp      eip: 144 eax:   4 ebp: 936 esp: 916 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax          eip: 148 eax:   4 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib            eip: 100 eax:   4 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   4 ebp: 936 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   4 ebp: 904 esp: 904 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   4 ebp: 904 esp: 900 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   4 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   4 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2              eip: 132 eax:   4 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax  eip: 136 eax:   4 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax       eip: 140 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp      eip: 144 eax:   3 ebp: 904 esp: 884 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax          eip: 148 eax:   3 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib            eip: 100 eax:   3 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   3 ebp: 904 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   3 ebp: 872 esp: 872 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   3 ebp: 872 esp: 868 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2              eip: 132 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax  eip: 136 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax       eip: 140 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp      eip: 144 eax:   2 ebp: 872 esp: 852 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax          eip: 148 eax:   2 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib            eip: 100 eax:   2 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   2 ebp: 872 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   2 ebp: 840 esp: 840 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   2 ebp: 840 esp: 836 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
jg .L2              eip: 132 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl 8(%ebp), %eax  eip: 136 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $1, %eax       eip: 140 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $12, %esp      eip: 144 eax:   1 ebp: 840 esp: 820 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %eax          eip: 148 eax:   1 ebp: 840 esp: 816 edx:   0 ebx:   0 ecx: 1004 flags:   0
call fib            eip: 100 eax:   1 ebp: 840 esp: 812 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   1 ebp: 840 esp: 808 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   1 ebp: 808 esp: 808 edx:   0 ebx:   0 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   1 ebp: 808 esp: 804 edx:   0 ebx:   0 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
jg .L2              eip: 124 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl $1, %eax       eip: 128 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
jmp .L3             eip: 188 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
movl -4(%ebp), %ebx eip: 192 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   0 ecx: 1004 flags:  64
leave               eip: 196 eax:   1 ebp: 840 esp: 812 edx:   0 ebx:   0 ecx: 1004 flags:  64
ret                 eip: 152 eax:   1 ebp: 840 esp: 816 edx:   0 ebx:   0 ecx: 1004 flags:  64
addl $16, %esp      eip: 156 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx     eip: 160 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl 8(%ebp), %eax  eip: 164 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $2, %eax       eip: 168 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:  64
subl $12, %esp      eip: 172 eax:   0 ebp: 840 esp: 820 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %eax          eip: 176 eax:   0 ebp: 840 esp: 816 edx:   0 ebx:   1 ecx: 1004 flags:   0
call fib            eip: 100 eax:   0 ebp: 840 esp: 812 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebp          eip: 104 eax:   0 ebp: 840 esp: 808 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl %esp, %ebp     eip: 108 eax:   0 ebp: 808 esp: 808 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebx          eip: 112 eax:   0 ebp: 808 esp: 804 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $4, %esp       eip: 116 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)    eip: 120 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags: 128
jg .L2              eip: 124 eax:   0 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl $1, %eax       eip: 128 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags: 128
jmp .L3             eip: 188 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl -4(%ebp), %ebx eip: 192 eax:   1 ebp: 808 esp: 800 edx:   0 ebx:   1 ecx: 1004 flags: 128
leave               eip: 196 eax:   1 ebp: 840 esp: 812 edx:   0 ebx:   1 ecx: 1004 flags: 128
ret                 eip: 180 eax:   1 ebp: 840 esp: 816 edx:   0 ebx:   1 ecx: 1004 flags: 128
```

```
addl $16, %esp       eip: 184 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   2 ebp: 840 esp: 832 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   2 ebp: 872 esp: 844 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 152 eax:   2 ebp: 872 esp: 848 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 156 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   1 ebp: 872 esp: 852 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   2 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 872 esp: 840 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 840 esp: 840 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 840 esp: 836 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   2 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   2 ecx: 1004 flags:  64
ret                  eip: 180 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   2 ecx: 1004 flags:  64
addl $16, %esp       eip: 184 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   3 ebp: 872 esp: 864 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   3 ebp: 904 esp: 876 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 152 eax:   3 ebp: 904 esp: 880 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 156 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   4 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   2 ebp: 904 esp: 884 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   3 ecx: 1004 flags:   0
call fib             eip: 100 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 904 esp: 872 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 872 esp: 872 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   2 ebp: 872 esp: 868 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   1 ebp: 872 esp: 852 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   3 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 872 esp: 840 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 840 esp: 840 edx:   0 ebx:   3 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 840 esp: 836 edx:   0 ebx:   3 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   3 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   3 ecx: 1004 flags:  64
ret                  eip: 152 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   3 ecx: 1004 flags:  64
addl $16, %esp       eip: 156 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:  64
subl $12, %esp       eip: 172 eax:   0 ebp: 872 esp: 852 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   0 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags:   0
call fib             eip: 100 eax:   0 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 872 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 840 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   0 ebp: 840 esp: 836 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jg .L2               eip: 124 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags: 128
ret                  eip: 180 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags: 128
addl $16, %esp       eip: 184 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   3 ecx: 1004 flags:   0
leave                eip: 196 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   3 ecx: 1004 flags:   0
```

```
ret                  eip: 180 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   3 ecx: 1004 flags:   0
addl $16, %esp       eip: 184 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   5 ebp: 904 esp: 896 edx:   0 ebx:   3 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   5 ebp: 904 esp: 896 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   5 ebp: 936 esp: 908 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 152 eax:   5 ebp: 936 esp: 912 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 156 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   5 ebp: 936 esp: 928 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   3 ebp: 936 esp: 916 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   3 ebp: 936 esp: 912 edx:   0 ebx:   5 ecx: 1004 flags:   0
call fib             eip: 100 eax:   3 ebp: 936 esp: 908 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   3 ebp: 936 esp: 904 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   3 ebp: 904 esp: 904 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   3 ebp: 904 esp: 900 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   2 ebp: 904 esp: 884 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   5 ecx: 1004 flags:   0
call fib             eip: 100 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   2 ebp: 904 esp: 872 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   2 ebp: 872 esp: 872 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   2 ebp: 872 esp: 868 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
jg .L2               eip: 132 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 136 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $1, %eax        eip: 140 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $12, %esp       eip: 144 eax:   1 ebp: 872 esp: 852 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %eax           eip: 148 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   5 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 872 esp: 840 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 840 esp: 840 edx:   0 ebx:   5 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 840 esp: 836 edx:   0 ebx:   5 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   5 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   5 ecx: 1004 flags:  64
ret                  eip: 152 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   5 ecx: 1004 flags:  64
addl $16, %esp       eip: 156 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   0 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:  64
subl $12, %esp       eip: 172 eax:   0 ebp: 872 esp: 852 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   0 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags:   0
call fib             eip: 100 eax:   0 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   0 ebp: 872 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   0 ebp: 840 esp: 840 edx:   0 ebx:   1 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   0 ebp: 840 esp: 836 edx:   0 ebx:   1 ecx: 1004 flags:   0
subl $4, %esp        eip: 116 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jg .L2               eip: 124 eax:   0 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl $1, %eax        eip: 128 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
jmp .L3              eip: 188 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 840 esp: 832 edx:   0 ebx:   1 ecx: 1004 flags: 128
leave                eip: 196 eax:   1 ebp: 872 esp: 844 edx:   0 ebx:   1 ecx: 1004 flags: 128
ret                  eip: 180 eax:   1 ebp: 872 esp: 848 edx:   0 ebx:   1 ecx: 1004 flags: 128
addl $16, %esp       eip: 184 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   1 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   2 ebp: 872 esp: 864 edx:   0 ebx:   5 ecx: 1004 flags:   0
leave                eip: 196 eax:   2 ebp: 904 esp: 876 edx:   0 ebx:   5 ecx: 1004 flags:   0
ret                  eip: 152 eax:   2 ebp: 904 esp: 880 edx:   0 ebx:   5 ecx: 1004 flags:   0
addl $16, %esp       eip: 156 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl %eax, %ebx      eip: 160 eax:   2 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl 8(%ebp), %eax   eip: 164 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $2, %eax        eip: 168 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
subl $12, %esp       eip: 172 eax:   1 ebp: 904 esp: 884 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %eax           eip: 176 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   2 ecx: 1004 flags:   0
call fib             eip: 100 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebp           eip: 104 eax:   1 ebp: 904 esp: 872 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl %esp, %ebp      eip: 108 eax:   1 ebp: 872 esp: 872 edx:   0 ebx:   2 ecx: 1004 flags:   0
pushl %ebx           eip: 112 eax:   1 ebp: 872 esp: 868 edx:   0 ebx:   2 ecx: 1004 flags:   0
```

```
subl $4, %esp        eip: 116 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:   0
cmpl $1, 8(%ebp)     eip: 120 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
jg .L2               eip: 124 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl $1, %eax        eip: 128 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
jmp .L3              eip: 188 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
movl -4(%ebp), %ebx  eip: 192 eax:   1 ebp: 872 esp: 864 edx:   0 ebx:   2 ecx: 1004 flags:  64
leave                eip: 196 eax:   1 ebp: 904 esp: 876 edx:   0 ebx:   2 ecx: 1004 flags:  64
ret                  eip: 180 eax:   1 ebp: 904 esp: 880 edx:   0 ebx:   2 ecx: 1004 flags:  64
addl $16, %esp       eip: 184 eax:   1 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   2 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   3 ebp: 904 esp: 896 edx:   0 ebx:   5 ecx: 1004 flags:   0
leave                eip: 196 eax:   3 ebp: 936 esp: 908 edx:   0 ebx:   5 ecx: 1004 flags:   0
ret                  eip: 180 eax:   3 ebp: 936 esp: 912 edx:   0 ebx:   5 ecx: 1004 flags:   0
addl $16, %esp       eip: 184 eax:   3 ebp: 936 esp: 928 edx:   0 ebx:   5 ecx: 1004 flags:   0
addl %ebx, %eax      eip: 188 eax:   8 ebp: 936 esp: 928 edx:   0 ebx:   5 ecx: 1004 flags:   0
movl -4(%ebp), %ebx  eip: 192 eax:   8 ebp: 936 esp: 928 edx:   0 ebx:   0 ecx: 1004 flags:   0
leave                eip: 196 eax:   8 ebp: 984 esp: 940 edx:   0 ebx:   0 ecx: 1004 flags:   0
ret                  eip: 264 eax:   8 ebp: 984 esp: 944 edx:   0 ebx:   0 ecx: 1004 flags:   0
addl $16, %esp       eip: 268 eax:   8 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl %eax, -20(%ebp) eip: 272 eax:   8 ebp: 984 esp: 960 edx:   0 ebx:   0 ecx: 1004 flags:   0
movl -16(%ebp), %edx eip: 276 eax:   8 ebp: 984 esp: 960 edx:   5 ebx:   0 ecx: 1004 flags:   0
movl -20(%ebp), %eax eip: 280 eax:   8 ebp: 984 esp: 960 edx:   5 ebx:   0 ecx: 1004 flags:   0
addl %eax, %edx      eip: 284 eax:   8 ebp: 984 esp: 960 edx:  13 ebx:   0 ecx: 1004 flags:   0
movl -12(%ebp), %eax eip: 288 eax:   4 ebp: 984 esp: 960 edx:  13 ebx:   0 ecx: 1004 flags:   0
addl %edx, %eax      eip: 292 eax:  17 ebp: 984 esp: 960 edx:  13 ebx:   0 ecx: 1004 flags:   0
movl -4(%ebp), %ecx  eip: 296 eax:  17 ebp: 984 esp: 960 edx:  13 ebx:   0 ecx: 1004 flags:   0
leave                eip: 300 eax:  17 ebp: 996 esp: 988 edx:  13 ebx:   0 ecx: 1004 flags:   0
leal -4(%ecx), %esp  eip: 304 eax:  17 ebp: 996 esp: 1000 edx:  13 ebx:   0 ecx: 1004 flags:   0
ret                  eip:   0 eax:  17 ebp: 996 esp: 1004 edx:  13 ebx:   0 ecx: 1004 flags:   0
[ssdavis@lect1 p5]$
```