CSCI2121 Assignment#7

James MacPhee B00768516

#1.

|  |  |
| --- | --- |
| **Instruction** | **Meaning** |
| movl $0, %eax | Move 0 into eax |
| andl $0x05, %ebx | Store 516 AND ebx |
| movl (%ecx), %edx | Move contents ecx into edx |
| movl -4(%ebp), %eax | Moves ebp-4 into eax |
| movl -4(%ebp, %edx, 4), %eax | Moves \*(ebp +(edx \*4) - 4) into eax |
| imull $50, %eax, %ebx | Store 50\*eax\*ebx in ebx |
| movb $0x05, al | Move 516 into al |

#2

main: pushl %rbp  
 movl %rsp, %rbp  
 movl -4(%rbp), %eax  
 cmpl %eax, -8(%rbp)   
 jge L2  
 movl -4(%rbp), %eax  
 subl -8(%rbp), %eax  
 movl %eax, -12(%rbp)  
 jmp L3  
 \_L2: movl -4(%rbp), %ebx  
 movl -8(%rbp), %eax  
 addl %ebx, %eax  
 movl %eax, -12(%rbp)  
 \_L3: movl $0, %eax  
 popl %rbp  
 ret

#3

|  |  |
| --- | --- |
| **Assembly Code** | **Meaning** |
| subl $16, %esp | Allocate space by moving up esp 16 bytes |
| movl $0, -4(%ebp) | Put 0 as value of first variable in stack |
| movl $10, -8(%ebp) | Put 10 as value of second variable in stack |
| movl $5, -12(%ebp) | Put 5 as value of third variable in stack |
| movl $0, -16(%ebp) | Put 0 as value of fourth variable in stack |
| \_L1: | L1 starts |
| movl -16(%ebp), %eax | Store the content at ebp -16 into eax |
| cmpl -12(%ebp), %eax | Compares contents of [ebp] -12 with contents of eax register |
| jge \_L2 | Jump if content at ebp-12 is greater than or equal to content of eax |
| movl -16(%ebp), %eax | Store the content at ebp -16 into eax |
| addl -8(%ebp), %eax | Add the contents of ebp-8 into eax |
| movl %eax, -8(%ebp) | Store eax in the variable at ebp-8 |
| movl -16(%ebp), %eax | Store the content at ebp -16 into eax |
| addl $1, %eax | Increment eax by 1 |
| movl %eax, -16(%ebp) | Store eax in the variable at ebp-16 |
| jmp \_L1 | Jump back to L1 |
| \_L2: | L2 starts |
| movl -4(%ebp), %eax | Store the content at ebp -4 into eax |
| addl $16, %esp | Nothing more to do, move down esp 16 bytes |

C code:

int main(){

int a = 0;

int b = 10;

int c = 5;

int d = 0;

while(c>=d){

b += d;

d++;

}

}