**Lab3 Preliminary**

CS224

Section No: 1

Fall 2019

Lab No. 3

Fatih Sevban Uyanık

21602486

**Q1)**

1. Convert the number - 77.125 (do not miss the minus sign) to IEEE 754 standard. Show how you obtain mantissa and exponent. Give the final answer in hex.

|  |  |  |
| --- | --- | --- |
| Sign(S) | Exponent(E) | Mantissa(M) |

Single Precision

S = 1

77 = 1001101

0.125 = 001

-77.125 = - 1001101.001 = -1.001101001 x 2^6

E = 6 + bias = 6 + 127 = 133 = 10000101

M= 001101001 00000000000000

--> 1 10000101 00110100100000000000000

**--> 0xC29A4000**

Double Precision

S=1

77 = 1001101

0.125 = 001

-77.125 = - 1001101.001 = -1.001101001 x 2^6

E = 6 + bias = 6 + 1023 = 1029 = 10000000101

M= 001101001 0000000000000000000000000000000000000000000

--> 1 10000000101 0011010010000000000000000000000000000000000000000000

--> 1100 0000 0101 0011 0100 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

**--> 0xC053480000000000**

1. Convert - 77.125 to a hypothetical floating point representation. Give the final answer in hex. For both cases show how you obtain mantissa and exponent.

|  |  |  |
| --- | --- | --- |
| Sign(S) | Exponent(E) | Mantissa(M) |

Single Precision

S = 1

77 = 1001101

0.125 = 001

-77.125 = - 1001101.001 = -1.001101001 x 2^6

E = 6 + bias = 6 + 120 = 126 = 01111110

M= 001101001 00000000000000

--> 1 01111110 00110100100000000000000

--> 1011 1111 0001 1010 0100 0000 0000 0000

**--> 0xBF1A4000**

Double Precision

S=1

77 = 1001101

0.125 = 001

-77.125 = - 1001101.001 = -1.001101001 x 2^6

E = 6 + bias = 6 + 1020 = 1026 = 10000000010

M= 001101001 0000000000000000000000000000000000000000000

--> 1 10000000010 0011010010000000000000000000000000000000000000000000

--> 1100 0000 0010 0011 0100 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

**--> 0xC023480000000000**

1. Consider the following 32 bit hexadecimal number 0xc1a00000. Assuming that it is a Floating point number give its decimal equivalent. Show how you obtain mantissa and exponent.

**0xC1A00000**

--> 1100 0001 1010 0000 0000 0000 0000 0000

--> 1 10000011 01000000000000000000000

S = 1

E = 131 - 127 = 4

M = 0.25

--> Result: -(1.01)b \* 2^4 = (-1,25) \* 16 = **-20**

**Q2)**

.data

number: .asciiz "312804589"

new\_line: .asciiz "\n"

.text

la $a0, number

jal sum\_recursive

move $t1, $v0

li $v0, 1

move $a0, $t1

syscall

# main

li $v0, 10

syscall

sum\_recursive:

addi $sp, $sp, -32

sw $ra, 0($sp)

sw $s0, 4($sp)

sw $s1, 8($sp)

sw $s2, 12($sp)

sw $s3, 16($sp)

sw $s4, 20($sp)

sw $s5, 24($sp)

sw $s6, 28($sp)

move $s0, $a0

lbu $s1, 0($s0)

# BASE CASE

bne $s1, $zero, jump\_1

move $v0, $zero

j jump\_2

jump\_1:

addi $a0, $s0, 1

jal sum\_recursive

# RECURSIVE CASE

lbu $s1, 0($s0)

addi $s1, $s1, -48

move $s2, $v0

add $s2, $s2, $s1

move $v0, $s2

jump\_2:

lw $ra, 0($sp)

lw $s0, 4($sp)

lw $s1, 8($sp)

lw $s2, 12($sp)

lw $s3, 16($sp)

lw $s4, 20($sp)

lw $s5, 24($sp)

lw $s6, 28($sp)

addi $sp, $sp, 32

jr $ra

Q3)

Delete\_x:

addi $sp, $sp, -32

sw $ra, 0($sp)

sw $s0, 4($sp)

sw $s1, 8($sp)

sw $s2, 12($sp)

sw $s3, 16($sp)

sw $s4, 20($sp)

sw $s5, 24($sp)

sw $s6, 28($sp)

move $s0, $a0

move $s5, $a0

move $s1, $a1

move $s6, $zero

for\_5: beq $s0, $zero, exit\_5

lw $s3, 4($s0)

bne $s3, $s1, jump\_3

lw $s3, 0($s0)

beq $s3, $zero exit\_5

lw $s4, 0($s3)

sw $s4, 0($s0)

sw $zero, 0($s3)

addi $t9, $t9, -1

addi $s6, $s6, 1

jump\_3:

lw $s0, 0($s0)

j for\_5

exit\_5:

move $v1, $s5

move $v0, $s6

lw $ra, 0($sp)

lw $s0, 4($sp)

lw $s1, 8($sp)

lw $s2, 12($sp)

lw $s3, 16($sp)

lw $s4, 20($sp)

lw $s5, 24($sp)

lw $s6, 28($sp)

addi $sp, $sp, 32

jr $ra