

Computer Science & Engineering Department

I.I.T Kharagpur

Compilers Lab

3rd year CSE: 5th Semester (Lab Quiz 1)

Time limit: 1.15 hour, (Time: 15.30-16.45)

Max Marks: 25

Date: 1st Sep, 2022

Answer all the questions

1. The assembly language code corresponding to a C program developed to compute the square root of a positive integer following Newton's method is given below. Please annotate the assembly language program (add comments). Add comments only to the numbered lines by mentioning the number. No need to write down the assembly code. Your comment should connect to the logic of the implementation.

(10 marks)

Line#	Instruction	Line#	Instruction
1	.file "LQ1.c"	28	.L2:
2	.section .rodata	29	movq -16(%rbp), %rax
3	.LC0:	30	movq %rax, -8(%rbp)
4	.string "Enter a +ve number: "	31	movsd -16(%rbp), %xmm0
5	.LC1:	32	mulsd -16(%rbp), %xmm0
6	.string "%lf"	33	movsd -24(%rbp), %xmm1
7	.LC6:	34	addsd %xmm0, %xmm1
8	.string "sqrt(%f) = %f\n"	35	movsd -16(%rbp), %xmm0
	.text	36	addsd %xmm0, %xmm0
9	.globl main	37	movapd %xmm1, %xmm2
	.type main, @function	38	divsd %xmm0, %xmm2
10	main:	39	movapd %xmm2, %xmm0
	.LFB0:	40	movsd %xmm0, -16(%rbp)
	.cfi_startproc	41	movsd -8(%rbp), %xmm0
11	pushq %rbp	42	subsd -16(%rbp), %xmm0

	.cfi_def_cfa_offset 16	43	divsd -16(%rbp), %xmm0
	.cfi_offset 6, -16	44	movsd .LC3(%rip), %xmm1
12	movq %rsp, %rbp	45	andpd %xmm1, %xmm0
	.cfi_def_cfa_register 6	46	movsd .LC4(%rip), %xmm1
13	subq \$32, %rsp	47	mulsd %xmm1, %xmm0
14	movl \$.LC0, %eax		ucomisd .LC5(%rip), %xmm0
15	movq %rax, %rdi		seta %al
16	movl \$0, %eax		testb %al, %al
17	call printf	48	jne .L2
18	movl \$.LC1, %eax	49	movsd -24(%rbp), %xmm0
19	leaq -24(%rbp), %rdx	50	movl \$.LC6, %eax
20	movq %rdx, %rsi	51	movsd -16(%rbp), %xmm1
21	movq %rax, %rdi	52	movq %rax, %rdi
22	movl \$0, %eax	53	movl \$2, %eax
23	call __isoc99_scanf	54	call printf
24	movsd -24(%rbp), %xmm0	55	movl \$0, %eax
25	movsd .LC2(%rip), %xmm1	56	leave
26	divsd %xmm1, %xmm0		.cfi_def_cfa 7, 8
27	movsd %xmm0, -16(%rbp)	57	Ret
			.cfi_endproc

2. Consider an input text file "student.txt", which contains records of all the students in a class. Each student record contains student name and student marks as "field name=field value" pairs, in a specific format as mentioned below, with one entry in each line.

Name=student name (say, xx)

Marks=total marks (say, yy)

where field value *student name* contains a **string of alphabets** (without space), and the *total marks* contain any **positive real numbers**. The field name(s) **Name** and **Marks** should be followed by = sign.

Write a lexical analyzer which scans the "student.txt" and prints the student information in the following format

"Name is xx and Marks is yy"

Your program should handle the following errors in the "student.txt"

- If *field values* such as "student name" and "total marks" violate the aforementioned format.
- If the = symbol is missing after the field names (Name and Marks)
The suitable error message with line numbers should be printed.
You may safely ignore any other kind of errors (say, errors in field name(s) etc)

You have to write the (a) Flex/Lex specifications, and the (b) C program, which implements the lexical analyzer.

Sample input/output(s)

(10+5 marks)

Example 1:

Student.txt

Name = Ram

Marks = 45

Name = Sham

Marks = 32

Name = Rohan

Marks = 55.7

Output :

Name is Ram and marks is 45

Name is Sham and marks is 32

Name is Rohan and marks is 55.7

20/11

Example 2 :

Student.txt

Name Ram

Marks = 45

Name = Sham

Marks = 32

Name = Rohan

Marks = 55

Output:

Syntax error in line 1, Expected a '=' but found Ram

Example 3 :

Student.txt

Name = Ram

Marks = 45

Name = 58

Marks = 32

Name = Rohan

Marks = 55

Output:

Name is Ram and marks is 45

Syntax error in line 3, Expected an identifier but found 58

Example 4 :

Student.txt

Name = Ram

Marks = 45

Name = Sham

Marks = thirty

Name = Rohan

Marks = 55

Output:

Name is Ram and marks is 45

Name is Sham Syntax error in line 4, Expected an integer but found thirty

*****END*****