Sl No	Answer
1	.LFB3: 01 .cfi_startproc # CFI Directive 02 pushq %rbp # Save old base pointer 03 .cfi_def_cfa_offset 16 # CFI Directive 04 .cfi_offset 6, -16 # CFI Directive 05 movq %rsp, %rbp # rbp <-- rsp set new base pointer 06 .cfi_def_cfa_register 6 # CFI Directive 07 movl %edi, -20(%rbp) # Store the value of n in stack at rbp -20 08 movq %rsi, -32(%rbp) # Store the address of the first matrix in stack at rbp -32 09 movq %rdx, -40(%rbp) # Store the address of the second matrix in stack at rbp -40 10 movl $0, -4(%rbp) # Set result to 0 11 movl $0, -8(%rbp) # Set i to 0 12 jmp .L26 # Jump to .L26 .L27: 13 movl -8(%rbp), %eax # Move i to eax 14 cltq # Convert integer type to 64 bit and store in rax 15 leaq 0(,%rax,4), %rdx # Load address 4*rax to rdx (rdx = 4*i) 16 movq -32(%rbp), %rax # rax = Mem[rbp-32] (rax = L) 17 addq %rdx, %rax # rax = rax+rdx (rax = L+4*i) 18 movl (%rax), %edx # Move value at rax to edx (edx = L[i]) 19 movl -8(%rbp), %eax # Move Mem[rbp-8](i) to eax 20 cltq # Convert integer to 64 bit and store in rax 21 leaq 0(,%rax,4), %rcx # Load address 4*rax to rcx (rdx = 4*i) 22 movq -40(%rbp), %rax # rax = Mem[rbp-40] (rax = R) 23 addq %rcx, %rax # rcx = rcx+rax (rcx = R+4*i) 24 movl (%rax), %eax # Move value at rax to eax (eax = R[i]) 25 imull %edx, %eax # eax = eax*edx (eax = L[i]*R[i]) 26 addl %eax, -4(%rbp) # result = result + eax (result+=L[i]*R[i]) 27 addl $1, -8(%rbp) # Increment i by 1 .L26: 28 movl -8(%rbp), %eax # Move i to eax 29 cmpl -20(%rbp), %eax # Compare n and i 30 jl .L27 # If (i<n) go to .L27 31 movl -4(%rbp), %eax # Move result to eax

	32 popq %rbp # Remove rbp from stack 33 .cfi_def_cfa 7, 8 # CFI Directive 34 ret # return 35 .cfi_endproc # CFI Directive NOTE: 0.5 marks for each line-wise correct comment.
2	%% “+” { printf(“ADD\n”); } “-” { printf(“SUBTRACT\n”); } “*” {printf(“MULTIPLY\n”); } “/” { printf(“DIVIDE\n”); } “|” { printf(“ABS\n”); } [0-9]+ { printf(“Number=%s\n”,yytext); } \n { printf(“Newline\n”); } [\t] {} . { printf(“Illegal input %s\n”,yytext);} %% NOTE: 8 marks for rule 1-5 and 2 marks for rule 6-9. Missing of any rule from 1-5 will incur penalty of 2 marks with a minimum, and maximum score from rule 1-5 is 0, and 8, respectively. Penalty (-2) for missing %%. Penalty (-2) if last rule goes above. Total marks for this question should not be negative.