Register Number | 1 | 9 | 5 | 0 | 0 | 1 | 0 | 7 | 6 |

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Department of Computer Science and Engineering

Continuous Assessment Test – III - Question Paper

Degree & Branch	**BE (CSE)**				**Semester**	VI
Subject Code & Name	**UCS1602 – Compiler Design**				**Regulation:**	**2018**
Academic Year	2021-2022	**Batch**	2019-2023	**Date**	**02-06-2022**	**FN**
Time: 90 Minutes **8.30 – 10.00 am**	**Answer All Questions**				**Maximum: 50 Marks**	

Part – A (6×2 = 12 Marks)

<KL1>	1.List the differences between syntax directed definition and translation scheme.	<CO3>

<KL3>	2.Apply the following SDT to predict the output for the input xxxxxxyzzz.
	S →xxW { print (6) }
	S → y { print(2) }
	W → Sz {print (3) }
		<CO3>

<KL3>	3. Calculate the total instruction cost for the given target code. MOV 6(R0),R1 ADD b,R1 SUB #3,R1 MOV R1, a ADD R0,R1 MOV R1, d	<CO4>
<KL2>	4. Compare register and address descriptors.	<CO4>
<KL2>	5. Define peephole optimization.	<CO5>

<KL3>	6. Identify the leaders in the following set of three address codes to build the basic blocks.
	if a < b goto L1
	goto L2
	L1: t1 = x + y
	t2= z * t1
	t3 = t1 + t2
	g=t3
	if c > d goto L2
	g=0
	L2: t1=x-y
	h=t1
		<CO5>

Part – B (3×6 = 18 Marks)

<KL2>	7. Explain syntax directed translation for assignment statements with example.	<CO3>
<KL2>	8. Explain the issues of code generation phase with example.	<CO4>
<KL2>	9. Explain code generation algorithm and generate assembly code for the following code. a=b*-c + b* -c	<CO4>

Part – C (2×10 = 20 Marks)

<KL3>	10. Apply SDT for flow of control statements to generate three address code for the following code.
	main()
	{
	int i, x=1,y=2,z=3,m,n;
	while(i<=10)
	{
	i=i+1;
	switch(i)
		<CO3>

```
            {
                    case 0:
                            m =x*z+y;
                            break;
                    case 2:
                            m=x/z+y;
                            break;
                    default:
                            n=x+z-y;
                                break;
            }
        }
    }
```

(OR)

<KL3>

11. Explain SDT for declaration statements and apply the same to construct the required symbol tables for the following nested procedures.

```
Procedure sort
   var a:array[1..n] of integer
   Procedure readarray
        var i:integer
   Procedure exchange(i,j:integer)
   Procedure quicksort(m,n:integer)
        var k,v : integer;
        function partition(x,y : integer):integer;
             var i,f : integer;
```

<CO3>

<KL3>

12. Explain the algorithm for Directed Acyclic Graph (DAG) generation from basic block and apply DAG to optimize the following code

```
1.  t1 : = 4 * i
2.  t2 : = a[t1]
3.  t3 : 4 * i
4.  t4 : = b[t3]
5.  t5:=4*i
6.  t6:=c[t5]
7.  t7 : = t2 * t4
8.  t8:=t7 +t6
9.  t9 : = p + t7
10. t10:=t8 +t9
11. p: = t10
12. t11 : = i + 1
13. i : = t11
14. if i < = 20 goto (1)
```

<CO5>

(OR)

<KL3>

13. Construct a flow graph for the following code after converting into three address code and do the necessary optimization.

```
a = 20;
c = a*35/3;
while (a<100)
{
        b = x * 1;
        if (a > b)
           a=a+30;
        else
           a=b+30;
}
```

<CO5>

40 x