SV ASSIGNMENT 2

1.

```
reg[1:0] array_s[0:10][0:256];
```

Using foreach

a. Initialize the above array with positive random values

b. Display all the locations contents

2.

```
module name;
reg[7:0] var1[1:10];
            initial
              begin
                        var1[0] = ’d10;
                        var1[1][4:0] = ’d31;
                        var1[2:10] = ‘d42;
              end

endmodule
```

Write the above program and simulate it.

3.

```
module name;
reg[31:0] mem [7:0][0:3];

initial
            begin
                        mem[0][0]=’d40;
                        mem[1][0:2]=’d52;
                        mem[1][3][5:0]=’d110;
            end
endmodule
```

Write the above program and simulate it.

4.

```
bit[3:0][7:0] var1;
bit[7:0]var2[1:0];
```

Initialize the above arrays var1 and var2 with any values.
Increment the values of var1 and var2 by 10 and display the result.

5.

```
bit[2:0][7:0] var1,var2;
bit[2:0][7:0] sum;
```

a.Initialize var1, var2. Add var1,var2 and store the result into sum array.

```
bit[7:0] var1[3:0],var2[3:0];
bit[7:0]sum[3:0];
```

b.Initialize var1, var2. Add var1,var2 and store the result into sum array.

6.
reg[31:0]array[7:0];
In the declaration itself, initialize the array with all 8 elements. If an error occurs, how to avoid it. Explore the reason .

7.
reg[31:0]array[7:0];
Initialize the above array with 5,10, 2, 8, 12, 50, 80. Store the >10 value elemets into another array. Do it in the simplest way.

8.
a. Initialize the dynamic array of size 20 elements with random values between 20 and 60.
Find out the indexes of the elements whose value is <50. Display those indexes.

b. Change the above dynamic array size to 30 elements and retain the old values.

9.
a. Take an array which should store the student names with respect to total marks

STUDENT_NAME	MARKS
Sai	90
Paul	70
Rao	79
Krish	75
Raone	95

b. After 10 time units check the name Krish is existed or not. If not then only store it as an element in the arrray otherwise modify the exisited element value by +10.

c. In the above array, find out the student name who got marks below 80.

10.
Declare an array with the below indexes and values can be random.
a.indexes 1stloc, 2ndloc, 3rdloc. Print the size of array.
b.indexes 100,200,5,40. Print the size of array.

11.Take an array with 20 students names and marks.

a.Then delete the students who got marks >90.

b.Find out how many students got marks <=90. Print the students names who got marks <=90.

12.
Declare an array with positive and negative indexes. Find out what are the first and last elements.

13.
array={“RAMA”:10, “RAJU”:20, “KING”:30, “KIND”:5};
Find out the first, next, last elements and how it got stored.

14.
a. int array1[int]={1:10,0:30};
 Add index 2 with value 5 and then print the array.
b. int q1[$]={1,10,20};
Insert an element 5 in between 1,10 in the queue.

15.
a.q1={10,40,5,100,20};
Delete the elements whose value is >20.(Get the indexes and deletes those indexes)
b. Declare a queue and initialize with random values between 0 to 50.
Delete the elements whose value is >20.(Get the indexes and deletes those indexes)

16.
q1={10,5,20};
a.Put 3 other values at front.
b.Put 4 other values at back.
c.Arrange all items in ascending order.
d. Delete last 4 elements.
e. Keep any 10 new elements at front one after other

17.
a. q1={-3,3,-6,4,2"};
Print the sum of all elements.
b. q2={4'b1010, 4'b1100, 4'b0101, 4'b0001};
Print the final XORed value of all elements.

18.
assoc={"ele1":10, "ele2":5, "el1":5, "el0":7, "e":10};
Store the unique values of above array into another array.

19.
a. string ass_arr[string];
ass_arr={"SUN":5, "SOON":10, "GOODS":2, "GOOD":20};
Find out first and last elements.

b. string q1[$];
q1={"SUN","SOON","GOODS","GOOD"};
Find out the first and last elements.

c. Check the difference between the above two outputs. Justify your answer.

20.

Write RTL code for 8 bit D flipflop.
It has two inputs, one is clock and other is [7:0]data_in.
It has one output, [7:0]data_out

a.In testbench take these arrays
int arr1[int]; //input array
int arr2[int];//output array
Store some random data into this array1.
Drive this array1 data on to the data_in line with respect to posedge clk.

Sample the data_out and store it in array2 with respect to negedge clk.

21.
Write RTL code for 8 bit D flipflop.
It has two inputs, one is clock and other is [7:0]data_in.
It has one output, [7:0]data_out

a.In testbench take these queues
int q1[$]; //input array
int q2[$];//output array
Store some random data into this q1.
Drive this q1 data on to the data_in line with respect to posedge clk.
Sample the data_out and store it in q2 with respect to negedge clk.

22.
Write a function that add two arrays and returns the result with return type array.

23.
Write a function that add two associative arrays and returns the result with return type associative array.

24.
Write a function that add two dynamic arrays and returns the result with return type dynamic array.

25.
Write a function that add two queues and returns the result with return type queue.

26.
enum{ORANCE,WHITE,GREEN}Flag;
Write a function that returns above enum type variable.
-Take one enum variable as input to the function and then modify the value of enumvarible with enumvarible.next() value in that function and return that enumvariable.