**­Experiment 2**

**Part 1:**

**Aim:**

1. Write the System Verilog code to­­:

1. Declare two bit-type fixed-size array of size 10 which can store 8bits values. Initialize first array with values {2,4,6,8,10,12,14,16,18,20} and second array with values {1,3,5,7,9,11,13,15,17,29}.
2. Display the values stored in two array declared in step-i.
3. Copy the contents of first array into second array and compare two arrays and display the result of comparison.

**Code:**

module arrays();

  bit [7:0] array1[10], array2[10];

  int x;

  initial begin

    array1 = '{2,4,6,8,10,12,14,16,18,20};

    array2 = '{1,3,5,7,9,11,13,15,17,29};

    $display("array1 is ",array1);

    $display("array2 is ",array2);

    array2 = array1;

    foreach(array1[i]) if(array1[i] != array2[i]) x = x+1;

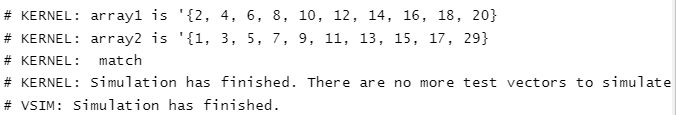
    if(x) $display("not match");

    else $display(" match");

  end

endmodule

**Output:**

****

**Part 2:**

**Aim:**

Write the SystemVerilog code to:

1. Declare the two integer type dynamic array.
2. Store the following values in the first dynamic array-{ 3,6,9,12,15,18} and {2,4,6,8,10,12,14} in the second dynamic array.
3. Print the sum of the elements stored in the arrays.
4. Insert the contents of the first array into second array (after its last element).
5. Delete the first array and try to print the contents of the first and second array. Write remark if you have any observation.

**Code:**

module arrays();

  integer array\_1[], array\_2[];

  int x,y;

  initial begin

    array\_1 = new[10];

    array\_1 = {2,4,6,8,10,12,14,16,18,20};

    array\_2 = new[10];

    array\_2 = {1,3,5,7,9,11,13,15,17,29};

    $display("array\_1 is ",array\_1);

    $display("array\_1 is ",array\_2);

    foreach(array\_1[i]) x = x+array\_1[i];

    $display("sum of elements in array\_1 is %0d",x);

    foreach(array\_2[i]) y = y+array\_2[i];

    $display("sum of elements in array\_2 is %0d",y);

    array\_2 = new[20]({array\_2,array\_1});

    array\_1.delete();

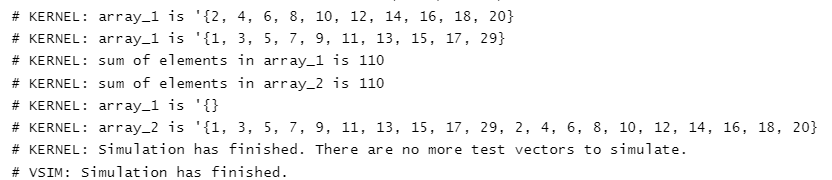
    $display("array\_1 is ",array\_1);

    $display("array\_2 is ",array\_2);

  end

endmodule

**Output:**

****

**Part 3:**

**Aim:**

Write the SystemVerilog code to:

1. Declare the two Queues Queue\_1 and Queue\_2. Initialize the first queue with values {3,4,5,6} and second queue with values {10,11,12,13}.
2. Insert 10 before 4 in queue\_1 and display the contents.
3. Insert Queue\_2 in Queue\_1 after the index value 3. Display the contents of the Queue\_1.
4. Delete value 3 from the first queue. Display the contents of the Queue\_1.
5. Insert value 20 at the front of Queue\_1. Display the contents of the Queue\_1.
6. Pop the last element of the Queue \_1. Display the contents of the Queue\_1.
7. Insert value 30 at the back of Queue\_1. Display the contents of the Queue\_1.
8. Pop the front element of the Queue \_1. Display the contents of the Queue\_1.

**Code:**

module arrays();

 int x;

  int queue\_1[$],queue\_2[$];

  initial begin

    int queue\_1[$] = {3,4,5,6};

    int queue\_2[$] = {10,11,12,13};

    $display("queue 1 is ",queue\_1);

    $display("queue 2 is ",queue\_2);

    queue\_1.insert(1,10);

    $display("queue 1 is ",queue\_1);

    foreach(queue\_2[i]) queue\_1.insert(4+i,queue\_2[i]);

    $display("queue 1 is ",queue\_1);

    queue\_1.delete(0);

    $display("queue 1 is ",queue\_1);

    queue\_1.push\_front(20);

    $display("queue 1 is ",queue\_1);

    x = queue\_1.pop\_back();

    $display("queue 1 is ",queue\_1);

    queue\_1.push\_back(30);

    $display("queue 1 is ",queue\_1);

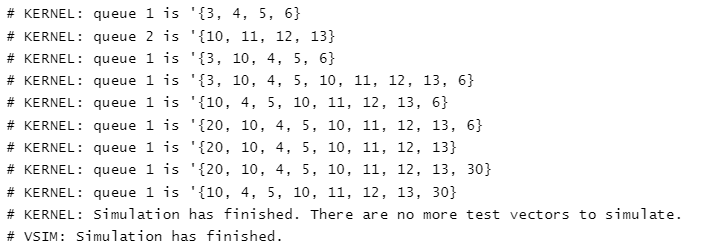
    x = queue\_1.pop\_front();

    $display("queue 1 is ",queue\_1);

  end

endmodule

**Output:**

****

**Result:**

The given problem statement is executed and verified to be correct.