<u>SYSTEM VERILOG</u>

CONSTRAINTS – Part 1

Write a constraint that generates odd numbers within 0 to 30.

```
class constraint_1;
      rand bit[4:0] a;
      constraint a_range{a % 2 == 1; a inside
{[10:30]};}
endclass

constraint_1 c1;

module test;
      initial
            begin
                  repeat(5)
                        begin
                              c1 = new();
                              assert(c1.randomize());
                              $display("a=%d",c1.a);
                        end
            end
 endmodule
```

Write a constraint to generate a pattern 1122334455.

```
class constraint_2;
      rand bit[2:0] da[];
      constraint c1{da.size == 10;}
      constraint c2{foreach(da[i])
                  da[i] == (i + 2) / 2;}
endclass

constraint_2 c1;

module test();
      initial
            begin
                  c1 = new;
                  assert(c1.randomize());
                  $display("da:%p",c1.da);
            end
endmodule
```

Write a constraint to generate a pattern 9753186420.

```
class constraint_3;
    rand int da[];
    constraint c1{da.size == 10;}
    constraint c2{foreach(da[i])
            if(i<5)
                da[i] == da.size - (i+(i+1));
            else
                da[i] == 18 - (i*2);}
endclass

constraint_3 c1;

module test();
    initial
        begin
            c1 = new;
            assert(c1.randomize());
            $display("da: %p",c1.da);
        end
endmodule
```

Write a constraint to generate a pattern 0011223344

```
class constraint_4;
      rand int da[];
      constraint c1{da.size == 10;}
      constraint c2{foreach(da[i])
                  da[i] == i/2;}
endclass

constraint_4 c1;

module test();
      initial
            begin
                  c1 = new;
                  assert(c1.randomize());
                  $display("da: %p",c1.da);
            end
endmodule
```

Write a constraint to generate a pattern 0101010101

```
class constraint_5;
    rand int da[];
    constraint c1{da.size == 10;}
    constraint c2{foreach(da[i])
            da[i] == i%2;}
endclass

constraint_5 c1;

module test();
    initial
        begin
            c1 = new;
            assert(c1.randomize());
            $display("da: %p",c1.da);
        end
endmodule
```

Write a constraint to generate a pattern 1010101010

```
class constraint_6;
    rand int da[];
    constraint c1{da.size == 10;}
    constraint c2{foreach(da[i])
            if(i%2 == 0)
                da[i] == 1;
            else
                da[i] == 0;}
endclass

constraint_6 c1;

module test();
    initial
        begin
            c1=new;
            assert(c1.randomize());
            $display("da: %p",c1.da);
        end
endmodule
```

Write a constraint to generate a pattern 2, 3, 5, 6, 8, 9, 11, 12, 14, 15

```
class constraint_7;
    rand int da[];
    constraint c1{da.size == 10;}
    constraint c2{foreach(da[i])
            if(i == 0)
                da[i] == 2;
            else if(i == 1)
                da[i] ==3;
            else if(i/2 == 0)
                da[i] == da[i-2] + 3;
            else if(i/2 != 0)
                da[i] == da[i-2] + 3;
            }
endclass
constraint_7 c1;
module test();
    initial
        begin
            c1=new;
            assert(c1.randomize());
            $display("da: %p",c1.da);
        end
endmodule
```

Write a constraint to generate a pattern 1234554321

```
class constraint_8;
      rand int da[];
      constraint c1{da.size == 10;}
      constraint c2{foreach(da[i])
                  if(i<5)
                        da[i] == i + 1;
                  else
                        da[i] == 10 - i;}
endclass
constraint_8 c1;
module test();
      initial
            begin
                  c1=new;
                  assert(c1.randomize());
                  $display("da: %p",c1.da);
            end
endmodule
```

Write a constraint to generate a pattern 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927

```
class constraint_9;
      rand int da[];
      constraint c1{da.size == 15;}
      constraint c2{foreach(da[i])
                  if(i<2)
                        da[i] == 0;
                  else if(i == 2)
                        da[i] == 1;
                  else
                        da[i] == da[i-3] + da[i-2] + da[i-
1];}
endclass

constraint_9 c1;

module test();
      initial
            begin
                  c1=new;
                  assert(c1.randomize());
                  $display("da: %p",c1.da);
            end
endmodule
```

Write a constraint to generate a pattern 9, 19, 29, 39, 49, 59, 69, 79, 89, 99

```
class constraint_10;
    rand int da[];
    constraint c1{da.size == 10;}
    constraint c2{foreach(da[i])
            da[i] == (i * 10) + 9;}
endclass

constraint_10 c1;

module test();
    initial
        begin
            c1=new;
            assert(c1.randomize());
            $display("da: %p",c1.da);
        end
endmodule
```