```
(*
inputs:
    r = r value in the text,
x0 = initialx value,
n = number  of iterations,
nreturn = number  of values to return, from  the end of the list generated
*)
run[r_, x0_, n_, nreturn_] := Module[{x}, x = Table[0, {n}];
    x[[1]] = x0;
    For[i = 1, i < n, i++, x[[i+1]] = r x[[i]] (1 - x[[i]])];
    Take[x, -nreturn]]

(*
new input:
      m  = number  of starting points
*)
runMultiple[r_, m_ , n_, nreturn_] :=
  Flatten@Table[run[r, x0, n, nreturn], {x0, 1 / (2 m ), 1, 1 / m }]

(* a sample  run *)
runMultiple[3.2, 3, 30, 4]

(*
The range of r values is specified by r1,r2,dr.
    The input called pr is the PlotRange
*)
rplot[r1_, r2_, dr_, pr_, n_, nreturn_] :=
  ListPlot[Transpose[Table[runMultiple[r, 10, n, nreturn], {r, r1, r2, dr}]],
    PlotMarkers → {Automatic , 2}, DataRange → {r1, r2},
    PlotRange → pr, AxesLabel → {"r", "x"}]

(* a test of the code.  Compare  this with prob 1. *)
Show[rplot[0, 4, 0.1, All, 1000, 100], ImageSize → 600]

(* Fig. 12.41 *)
Show[rplot[2.8, 4, 0.001, All, 1000, 100], ImageSize → 1200]

(* Fig. 12.44
  Note the first bifurcation is not well
  resolved here.  More iterations would be better. *)
Show[rplot[3.84, 3.856, 0.000014, {0.44, 0.56}, 1000, 100], ImageSize → 1200]

(* 2000 iterations.  The first bifurcation looks better,
but more  iterations would be nice. *)
Show[rplot[3.84, 3.856, 0.000014, {0.44, 0.56}, 2000, 100], ImageSize → 1200]
```