**8.1**

**Q10.**

**a) Find a recurrence relation for the number of ternary strings of length *n* that contain two**

**consecutive 0s.**

Let an be the number of ternary strings that contain two consecutive 0’s.

To construct such a string we could start with either a 1 or a 2 and follow with a string containing two consecutive 0’s (and this can be 198 Chapter 8 Advanced Counting Techniques done in 2an−1 ways), or we could start with 01 or 02 and follow with a string containing two consecutive 0’s (and this can be done in 2an−2 ways), we could start with 00 and follow with any ternary string of length n − 2 (of which thereare clearly ). Therefore the recurrence relation, valid for all n ³ 2, is

an = 2an−1 + 2an−2 +

**b) What are the initial conditions?**

Clearly a0 = a1 = 0.

**c) How many ternary strings of length six contain two**

**consecutive 0s?**

We will compute a2 through a6 using the recurrence relation:

a2 = 2a1 + 2a0 + 30 = 2 · 0 + 2 · 0 + 1 = 1

a3 = 2a2 + 2a1 + 31 = 2 · 1 + 2 · 0 + 3 = 5

a4 = 2a3 + 2a2 + 32 = 2 · 5 + 2 · 1 + 9 = 21

a5 = 2a4 + 2a3 + 33 = 2 · 21 + 2 · 5 + 27 = 79

a6 = 2a5 + 2a4 + 34 = 2 · 79 + 2 · 21 + 81 = 281

Thus there are 281 bit strings of length 6 containing two consecutive 0’s.

**Q20. Show that the Fibonacci numbers satisfy the recurrence relation fn = 5fn−4 + 3fn−5 for n = 5, 6, 7, . . . , together with the initial conditions f0 = 0, f1 = 1, f2 = 1, f3 = 2, and f4 = 3. Use this recurrence relation to show that f5n is divisible by 5, for n = 1, 2, 3, . . . .**

We let an be the number of ways to pay a toll of 5n cents.

(Obviously there is no way to pay a toll that isnot a multiple of 5 cents.)

the answer is : an = an−1+an−2 , with a0 = a1 = 1. So we find that a9 = 55.

**Q32.** Let {an} be a sequence of real numbers. The **backward differences** of this sequence are defined recursively as shown next. The **first difference** ∇an is ∇an = an − an−1. The **(k + 1)st difference** ∇k+1an is obtained from ∇kan by ∇k+1an = ∇kan −∇kan−1.

**Find ∇an for the sequence {an}, where**

**a) an = 4.**

**∇**an = 4 − 4 = 0

**b) an = 2n.**

**∇**an = 2n − 2(n − 1) = 2

**c) an = n2.**

∇an = − = 2n – 1

**d) an = 2n.**

∇an = − =