Ex 7

1. Algorithm to check if number is palindrome

Step 1. Input number (1234321)

Step 2. Declare rev, lastdigit, num

Step 3. WHILE (num != 0):

lastdigit = (num % 10)

rev = (rev * 10) + lastdigit

num = num / 10

Step 4. If (num == rev)

print ("number is palindrome")

Step 5. Else

print not palindrome.

Step 6. END.

2. Apply algorithm for 1234321

(i). num = 1234321

(ii). rev = 0, lastdigit = 0

(iii). While (~~123~~ num ≠ 0):

- lastgit = (1234321 % 10)
  = 1

- rev = (0x10) + 1
  = 1

- num = 1234321 / 10 = 123432

→ num $\neq 0$ $(123432 \neq 0)$

* lastd. $= (123432 \ \% \ 10)$

  $= 2$

* rev $= (1 \times 10) + 2$

  $= 12$

* num $= 123432 / 10 = 12343$

→ num $\neq 0$ $(12343 \neq 0)$

* lastd. $= (12343 \ \% \ 10)$

  $= 3$

* rev $= (12 \times 10) + 3$

  $= 123$

* num $= 12343 / 10 = 1234$

→ num $\neq 0$ $(1234 \neq 0)$

* lastd. $= (1234 \ \% \ 10)$

  $= 4$

* rev $= (123 \times 10) + 4$

  $= 1234$

* num $= 1234 / 10 = 123$

→ num $\neq 0$ $(123 \neq 0)$

* lastd. $= (123 \ \% \ 10)$

  $= 3$

* rev $= (1234 \times 10) + 3$

  $= 12343$

* num $= (123 / 10) = 12$

→ num ≠ 0 (12 ≠ 0)

* last d. = (12 % 10)

= 2

* rev = (12343 × 10) + 2

= 123432

* num = 12 / 10 = 1

→ num ≠ 0 (1 ≠ 0)

* last d. = (1 % 10)

= 1

* rev = (123432 × 10) + 1

= 1234321

* num = 1 / 10 = 0

→ num = 0 // terminate while loop

(iv). (rev == num)

(1234321 == 1234321) True

⇒ 1234321 is palindrome

(v). End.

EX 8 :

a) <u>Using Euclidean Algorithm to find GCD of 8256 and 3168</u>

Step 1. Begin

Step 2. Declare variables :
- * remainder as rem
- * numerator as num
- * denominator as den
- * gcd as gcd

Step 3. Initialize variables from step 2.

num = 8256
den = 3168

Step 4. If (num == 0 || den == 0):

return gcd = 1

STOP.

Step 5. Else :

rem = (num % den)
= (8256 % 3168)
rem = 1920

Step 6. WHILE (rem ≠ 0)

num = den ; ⇒ num = 3168
den = rem ; ⇒ den = 1920
rem = (num % den)
= (3168 % 1920)
rem = 1248

⊛ Since, rem $\neq$ 0 ; (rem = 1248)

num = den; $\Rightarrow$ num = 1920

den = rem; $\Rightarrow$ den = 1248

rem = (num % den)

= (1920 % 1248)

rem = 672

⊛ Since, rem $\neq$ 0 (rem = 672)

num = den ; $\Rightarrow$ num = 1248

den = rem ; $\Rightarrow$ den = 672

rem = (num % den)

= (1248 % 672)

rem = 576

⊛. Since, rem $\neq$ 0 (rem = 576)

num = den; $\Rightarrow$ num = 672

den = rem; $\Rightarrow$ den = 576

rem = (num % den)

= (672 % 576)

rem = 96

⊛. Since, rem $\neq$ 0 (rem = 96)

num = den ; num = 576

den = rem ; den = 96

rem = (num % den)

= (576 % 96)

rem = 0.

⊛. Since, rem = 0 → stop the iteration.

Step 7. End While

Step 8. gcd = den // since the last loop
den = rem = 96

gcd = 96

Step 9. Display gcd of num and den

⇒ The gcd of 8256 and 3168 is 96.

b) <u>Using GCD to find LCM of 8256 and 3168</u>

$$\text{Lcm} = (a * b) / \text{gcd} \quad \left(\frac{\text{num} * \text{den}}{\text{gcd}} \right)$$

$$= \frac{8256 * 3168}{96}$$

⇒ LCM = 272,448

The LCM of 8256 and 3168 is 272,448

Ex 9.

(a) and (b) **Generating first 12 terms of fibonacci**

Step 1. Enter $n = 12$

Step 2. Initialize array of size n

$\rightarrow$ fib[n] $\Rightarrow$ fib[12]

Step 3. Set:

fib[0] = 0

fib[1] = 1

Step 4. loop / Iterate through the array while adding other element.

$\hookrightarrow$ For $i = 2$ To $n = (\overset{11}{\overline{12-1}})$ Do:

fib[i] = fib[i-1] + fib[i+2]

⊗ i=2, fib[2] = 0 + 1 = 1

⊗ i=3, fib[3] = 1 + 1 = 2

⊗ i=4, fib[4] = 2 + 1 = 3

⊗ i=5, fib[5] = 3 + 2 = 5

⊗ i=6, fib[6] = 5 + 3 = 8

⊗ i=7, fib[7] = 8 + 5 = 13

⊗ i=8, fib[8] = 13 + 8 = 21

⊗ i=9, fib[9] = 21 + 13 = 34

⊗ i=10, fib[10] = 34 + 21 = 55

⊗ i=11, fib[11] = 55 + 34 = 89

Step 5. Display the fib sequences

0	1	1	2	3	5	8	13	21	34	55	89

Step 6. END.

~~8256 − 6336 = 1920~~

~~3168 | 2~~