ADS- Assignment-1
By: Muskan Gupta
2K19/SE/077

Ans1

1) Start
2) Declare variables n, i, flag
3) Initialize variables flag=1, i=2
4) Read n from user
5) If n<=1

   Display "Prime"

   ~~Goto~~ stop
6) For i< [(n/2)+1]

   If (n%i ==0)

   ~~brk~~ flag=0;
   break;

   i=i+1
7) If flag==0

   Display "Prime"

   Else

   Display "Not Prime"
8) Stop

Ans2 1) Start
2) Declare variables A, B, C, n, i
3) Initialize A=0, B=~~0~~1, i=2
4) ~~Display A, B~~

4~~5~~) ~~C=A+B~~ ~~for~~ Read n from user 5) Display A, B

6) for i<n

   C=A+B. Display C
   A=B
   B=C
   i=i+1

Muskan Gupta
2K19/SE/077

7) Stop

Ans3 1) Start

2) Declare array A[ ], n, i, x, flag

3) Input A[ ], x, n & initialise i = 0, flag = 0

4) for i < n

    if (A[i] == x)

        Display "x found at index i"

    ~~else~~ flag = 1

5) If flag == 0

    Display x not found

6) Stop.

Ans4 1) Start

2) Declare array A[ ], n, l, u, mid, x

3) Input A[ ], x, n & initialise l = 0, u = n-1, mid = (l+u)/2

4) while (l < u)

    mid = (l+u)/2

    if (A[mid] == x)

        Display "x found at index mid"

        STOP

    else if (A[mid] < x)

        l = mid + 1

    else

        u = mid - 1

   end while

5) STOP

Muskan Gupta
2K19/SE/077

Ans5 The problem can be ~~solved~~ formally stated as:-

Input: 2 binary integers of n-bit are stored in 2 n element array of binary digits (either 0 or 1) $A = \{a_1, a_2, a_3 \ldots a_n\}$ & $B = \{b_1, b_2, b_3 \ldots b_n\}$

Output: A (n+1) bit binary integer stored in (n+1) element array of binary digits (either 0 or 1) $C = \{c_1, c_2, c_3 \ldots c_n\}$ such that $A+B=C$

Pseudo code : ADD-BINARY(A, B)

1. n = Max(A.length, B.length)
2. Let C[n+1] be new array
3. carry = 0
4. for i = 1 to n
   C[i] = (A[i] + B[i] + carry) mod 2
   carry = ⌊(A[i] + B[i] + carry)/2⌋
5. C[n+1] = carry
6. return C
7. Stop

Ans6 A/c to question:-

$100n^2 < 2^n$

$n=1 \Rightarrow 100 > 2^1$
$n=2 \Rightarrow 400 > 2^2$
$=4 \Rightarrow 1600 > 2^4$
$=8 \Rightarrow 6400 > 2^8$
$=16 \Rightarrow 25600 < 2^{16}$
$=15 \Rightarrow 22500 < 2^{15}$

~~∴ for n=15 100~~

~~∴ So at n=15, A starts to run~~

∴ So at n=15, an algo whose running time is $100n^2$ runs faster

Muskan Gupta ~~century~~
2K19/SE/077

Ans 7	1 sec	1 min	1 hour	1 ~~month~~ day	1 ~~year~~ month	1 year ~~century~~	1 century
$\log n$	2^{10^6}	$2^{6 \cdot 10^7}$	$2^{36 \cdot 10^8}$	$2^{864 \cdot 10^8}$	$2^{25920 \cdot 10^8}$	$2^{315360 \cdot 10^8}$	$2^{31556736 \cdot 10^8}$
$\sqrt{n}$	10^{12}	$36 \cdot 10^{14}$	$1296 \cdot 10^{16}$	$746496 \cdot 10^{16}$	$6718464 \cdot 10^{18}$	$994519296 \cdot 10^{18}$	$995827586973696 \cdot 10^{16}$
n	10^6	$6 \cdot 10^7$	$36 \cdot 10^8$	$864 \cdot 10^8$	$2592 \cdot 10^9$	$31536 \cdot 10^9$	$31556736 \cdot 10^8$
$n \log n$	62746	2801417	133378058	2755147203	71870856409	797633893349	68654697441062
n^2	1000	7745	6000	293938	1609968	5615692	56175382
n^3	100	391	1532	4420	13736	31593	146677
2^n	19	25	31	36	41	44	51
$n!$	9	11	12	13	15	16	17