Arshita
2111024

DATE / / PAGE 1

Assignment - 1
EEN 103

(1)

Octal		Hexadecimal	
$16_{(10)}$	$20_{(8)}$	$16_{(10)}$	$10_{(16)}$
17	21	17	11
18	22	18	12
19	23	19	13
20	24	20	14
21	25	21	15
22	26	22	16
23	27	23	17
24	30	24	18
25	31	25	19
26	32	26	1A
27	33	27	1B
28	34	28	1C
29	35	29	1D
30	36	30	1E
31	37	31	1F
32	40	32	20

∴ for base 13, list of numbers from 8 to 28.

(10)	(13)	(10)	(13)	(10)	(13)
8	8	17	14	26	20
9	9	18	15	27	21
10	10	19	16	28	22
11	E	20	1A		
12	F	21	1B		
13	10	22	1C		
14	11	23	1D		
15	12	24	1E		
16	13	25	1F		

(2) (a) $14/2 = 5$

Let base = a

$1(a)^1 + 4(a)^0 = 5(a)^0$

$2(a)^0$

$a + 4 = 10$

$a = 6$

$\therefore$ Base = 6

(b) $\frac{54}{4} = 13$

Let base = a

$5(a)^1 + 4(a)^0 = 1(a)^1 + 3(a)^0$

$4(a)^0$

$5a + 4 = 4a + 12$

$a = 8$

$\therefore$ Base = 8

(c) $24 + 17 = 40$

Let base = a

$(2a + 4) + (a + 7) = 4a$

$a = 11$

$\therefore$ Base = 11

(3)

(a) N = B2FA

base = 16

(Radix Complement)

n = 4 r = 16

Comple. = $r^n - N$

= $(FFFF)_{16} - (B2FA)$

→ (4 D 0 6) 16's compliment

B

(15 − 11) = 4

(15 − 2) = 13 (D)

15 − 15 = 0

16 − 10 = 6

A

complement of A wrt 16, A is at unit place

(b) N = B2FA

A = 10 ↳ (11) (2) (15) (10)

B = 11

F = 15

Binary: 1011 0010 1111 1010

(c) 1011 0010 1111 1010

1's complement = 0100 1101 0000 0101

+ 1

2's complement 0100 1101 0000 0110

(d) 0001 0001 0000 0110
$2^3\ 2^2\ 2^1\ 2^0$
↳ (1) (1) (0) (6)

3 (c) 1011 0010 1111 1010

1's complement = 0100 1101 0000 0101
+ 1
0100 1101 0000 0110

(d) 0100 1101 0000 0110
(4) (13) (0) (6)

⇒ 4D06 → same as 3(a)

---

4 (a) 10011 − 10001
2's complement of 10001 is 01111

⇒ A − B = A + (−B)
↑
2's complement of B.

10011
+ 01111
Ans 00010

(b) 100010 − 100011
2's complement of 100011 ⇒ 011101

100010
011101
111111 → negative number (2's & affix -ve sign)

↳ − 000001
↓ +1
Ans = 111110 = −111110

(c) 1001 − 101000

2's complement of 101000 ⇒
$$\begin{array}{r} 010111 \\ +\quad 1 \\ \hline 011000 \end{array}$$

$$\begin{array}{r} 001001 \\ +\ 011000 \\ \hline 100001 \end{array}$$
→ negative number

2' complement = $-(01111\underset{+1}{0}) \rightarrow \boxed{-11111}$

(d) 110000 − 01010

2's complement of 01010 ⇒
$$\begin{array}{r} 10101 \\ +\quad 1 \\ \hline 10110 \end{array}$$

$$\begin{array}{r} 110000 \\ +\ 001011 \\ \hline 111011 \end{array}$$
→ negative

~~2's complement~~ ~~111011~~ ~~→ 000100~~
$$\begin{array}{r} +\quad 1 \\ \hline 101 \end{array}$$

→ By removing carry

⇒ 011011

---

(5) (a) signed 10's. * 10's complement of 09286

(a) (+9,286) + (+801)
= 09286 + 00801
= 10087

= AAAAA
− 09286
90714

= 99999
− 09286
90713
+ 1
90714

(b) $(+09286) + (-00801)$

↳ 10's complement

$$\begin{array}{r} 09286 \\ +\ 99199 \\ \hline 108485 \\ \hline \end{array}$$

↑

$$\begin{array}{r} 99999 \\ 00801 \\ \hline 99198 \\ +\quad 1 \\ \hline 99199 \\ \hline \end{array}$$

The final carry is 1 → remove

∴ Ans = 8485.

(c) $-09286 + 00801$

↓

10's complement 90714

$$\begin{array}{r} 90714 \\ 00801 \\ \hline 91515 \\ \hline \end{array}$$

as 09286 is large wrt 801, ans is negative.

10's complement

$$\begin{array}{r} 99999 \\ 91515 \\ \hline 8484 \\ +\ 1 \\ \hline -8485 \\ \hline \end{array}$$

$-09286 + 00801 = -8485.$

(d) $(-09286) + (-00801)$

↓ ↓

90714 99199

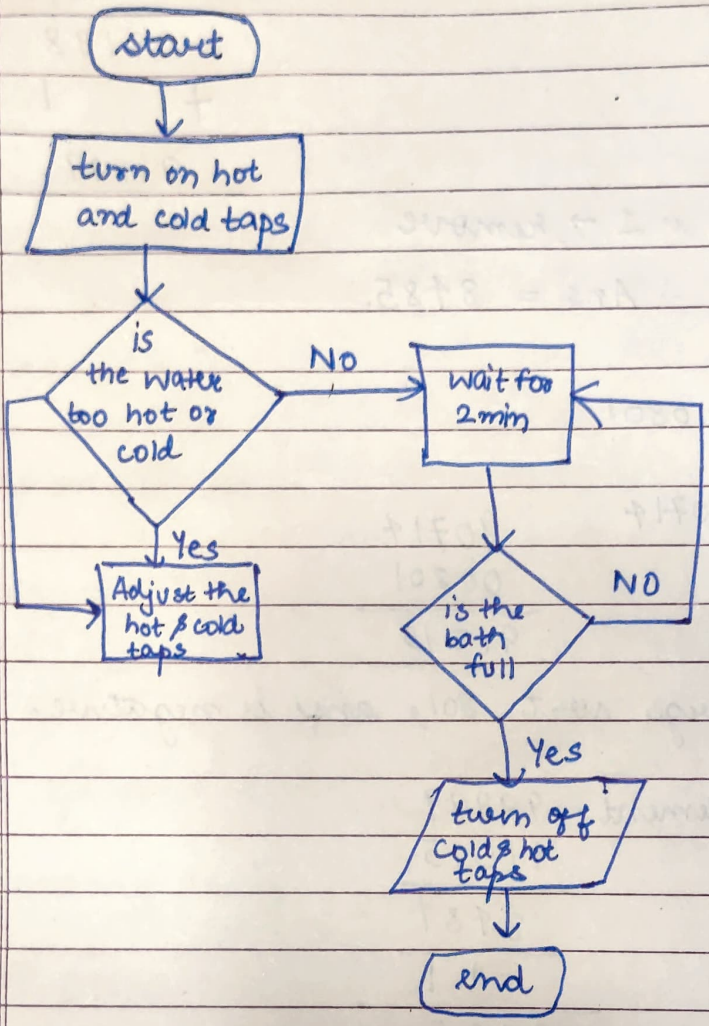
$$\begin{array}{r} 90714 \\ 99199 \\ \hline 189913 \\ \hline \end{array}$$

→ removing carry

89913 →

$$\begin{array}{r} 99999 \\ 89913 \\ \hline 10086 \\ +\quad 1 \\ \hline (-10087)\ \text{Ans} \\ \hline \end{array}$$

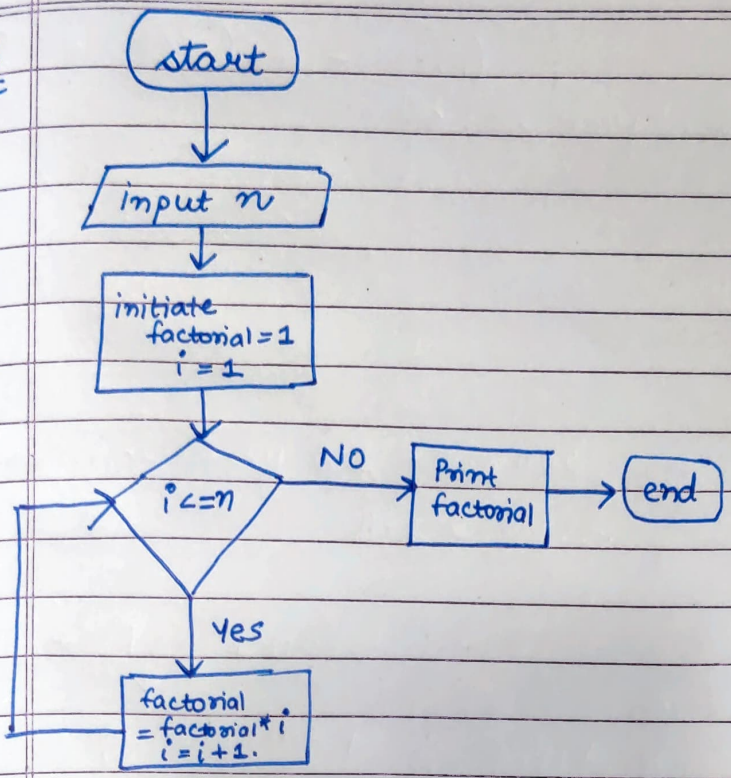
(6) To construct a flow chart for the process of filling a bath tub with water which involves →



(8) Algorithm to write factorial

Step 1: Start
Step 2: Input 1
Step 3: Declare / initiate factorial = 1 and i = 1
Step 4: check if i ≤ n
Step 5: Yes, factorial = factorial * i, NO: Stop.
Step 6: Repeat
Step 7: Display factorial
Step 8: End.

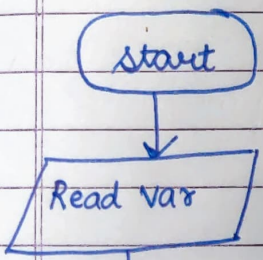
8



---

(7) • Terminations

Arrows → used to show flow of program



start — used to denote the start point of the program.

Read var — Used for taking input from the user and store it in 'var'.

Decision — Used to make decision(s) in the program and answer in form of True (Yes) or False (NO)

Yes/NO

Process — Used to perform operations in a program

end — used to denote end point of program.