C PROGRAM (EVALUATED BY AD)								
#	**Roll No**	**Name**	**Output (5)**	**Buildtree (10)**	**Populate Sizes (5)**	**printPC (5)**	**Total (25)**	**Comments**
1	18CS60D01	Abdul Shamnar P	0	8	0	0	**8**	Incomplete code. BuildTree() logic Okish. PopulateSizes() and printPC() not implemented.
2	18CS60D02	Amit Kumar Jha	0	7	0	0	**7**	Incomplete code. Only BuildTree() attempted, but it is not compilable. For example, n undeclared. Logic looks OK.
3	18CS60D03	Bachau Prasad	0	2	3	2	**7**	Incomplete code. No effort to BuildTree() as per question. Nodecount() written as populatesizes() is logically OK. Preorder() is incomplete implementation of PrintPC().
4	18CS60D04	Ajay Kumar Gupta	0	2	0	1	**3**	Practically nothing is done.
5	18CS60D05	Aditya Kaushik	0	2	0	0	**2**	Practically nothing is done.
6	18CS60D06	Anish Poonia	0	2	3	2	**7**	Practically nothing is done. PopulateSizes() is logically OK, but missing termination cases. PrintPreorder() is incomplete implementation of PrintPC().
7	18CS60R01	Soham Poddar	5	10	5	5	**25**	Perfect.
8	18CS60R02	Shounak Paul	5	9	5	5	**24**	Good. Include stdlib.h. There was no need to copy parts of pre[] and in[] to pre_l[], in_l[], pre_r[], in_r[].
9	18CS60R03	Neha	3	8	0	0	**11**	Lousy way of reading files. Read integers as integers. If n is 20, it reads n = 2. PopulateSizes() and PrintPC() not implemented.
10	18CS60R05	Himanshu Verma	0	2	0	2	**4**	Practically nothing is done.

11	18CS60R07	Souradip Guha	3	9	0	0	**12**	BuildTree() looks quite correct, but minor bugs may be there, because the output sequence does not match the input. PopulateSizes() and PrintPC() not implemented.
12	18CS60R08	Animesh Panwar	2	5	0	0	**7**	BuildTree() conceptually OK, but the implementation is seriously buggy. PopulateSizes() and PrintPC() not implemented.
13	18CS60R10	Manish Kumar	0	0	2	0	**2**	Practically nothing is done. C syntax is quite poor.
14	18CS60R11	Ahwan Mishra	0	3	2	0	**5**	BuildTree() uses only one array, why? PopulateSizes: You are not supposed to store left_size and right_size in a node. Reading is character-based.
15	18CS60R12	Sugandh Pargal	2	5	2	3	**12**	Very faulty. You are not supposed to store sizel, sizer in nodes. No effort of reading from file.
16	18CS60R14	Sharad Shourya Ghosh	0	2	0	0	**2**	No effort to solve the given problem.
17	18CS60R15	Gourav Chaturvedi	5	9	4	5	**23**	Do not use 0 to initialize pointers. PopulateSizes() and NodeCount() do essentially the same thing---inefficient to use both.
18	18CS60R16	Erudakar Omkar Vasant	2	10	5	0	**17**	File reading is faulty. PrintPC() not implemented.
19	18CS60R17	Udit Agrawal	0	7	0	0	**7**	BuildTree() conceptually OK, but the implementation is seriously buggy. PopulateSizes() and PrintPC() not implemented.
20	18CS60R18	Shivangi Sharma	5	10	4	5	**24**	Inefficient use of two functions count() and populatesizes().
21	18CS60R19	Rajesh Sahu	0	2	0	0	**2**	Practically nothing is done.
22	18CS60R20	Abhijeet Bhandari	0	4	0	0	**4**	This is not the algorithm to implement for buidTree(). No effort to read from file.

23	18CS60R21	Subham Saha	0	0	0	0	**0**	Nothing is done. Even reading data is faulty. No effort to even declare a tree node.
24	18CS60R22	Akashdeep	0	2	0	2	**4**	Practically nothing is done.
25	18CS60R23	Birudaraju Sri Charan	0	3	2	3	**8**	BuildTree() very faulty. You are not supposed to maintain sizel, sizer in each node. Why 1+ in populatesizes()?
26	18CS60R24	Aditi Singh	0	8	5	5	**18**	No output (program hangs). BuildTree() logic not clear. PopulateSizes() and PrintPC() codes are logically correct.
27	18CS60R25	Nidamanuri Dharma Teja	2	10	0	0	**12**	Seg Fault in BuildTree(). The function is logically OK. The last two functions not implemented.
28	18CS60R26	Ayush Malik	0	3	1	0	**4**	Invalid C syntax. No algorithm is properly implemented.
29	18CS60R28	Shivansh Gupta	1	8	5	5	**19**	Program logically OK, but encounters SegFault. No effort to read from file.
30	18CS60R30	Karwa Prachi Mukesh	3	8	3	3	**17**	Incomplete output. No effort of reading from file. IN[] and POST[] arrays are copied for children – not needed. You are not asked to maintain child sizes (not printed).
31	18CS60R31	Bansi Shah	3	7	3	3	**16**	Output buggy. BuildTree() make copies of IN[] and POST[] for subtrees. PopulateSizes() does not populate anything. PrintPC() does not print sizes. No effort to read from file.
32	18CS60R32	Subrata Chattopadhyay	1	5	5	3	**14**	Output: Seg Fault. BuildTree(): seriously buggy (why every invocation gets the same array?) No effort to read from file. Why should PrintPC() call PopulaateSizes()?
33	18CS60R34	Vaibhav Mishra	0	4	0	0	**4**	This is not the algorithm for BuildTree(). How come you build from only postorder traversal (it is not BST)? Populatesizes() and PrintPC() not implemented.
34	18CS60R35	Vishal Kumar	0	4	0	0	**4**	Very little done toward implementing the actual algorithm. Char-based file read is bad. Why are left and right subtrees set in a loop?

35	18CS60R36	Mayank Jain	0	3	0	0	**3**	Only left and right subtree traversals stored in arrays. No effort to build the tree actually. Not even a node data type declared.
36	18CS60R37	Surabhi S Kadur	0	2	0	0	**2**	Nothing is implemented except for a super-complicated way of reading the input file. Indentation is horrible.
37	18CS60R38	Doshi Rushi Kamaleshbhai	0	2	0	2	**4**	Extremely buggy. Constructtree() should return a pointer. Both recursive calls are on noOfNodes-1.
38	18CS60R39	Stanchion Bishoyi	0	0	0	0	**0**	No file submitted.
39	18CS60R40	Harshita Chouhan	0	2	0	0	**2**	Nothing is implemented except for a faulty char-based reading of the input file.
40	18CS60R41	Saket Kumar	0	7	0	0	**7**	Only some effort is made to construct the tree. The recursive call on sub2 should not be on post. Encounters Seg Fault. No file input.
41	18CS60R42	Gaurav Gupta	0	2	0	0	**2**	Nothing appropriate is implemented. Even scanf()'s do not have &. Indentation is horrible.
42	18CS60R43	Pranjal Doshi	3	4	3	4	**14**	Some effort is made, but all implemented algorithms are faulty. (Why +1 in printPC()? Should be stored in size.)
43	18CS60R44	Amit Kumar	1	0	0	0	**1**	Only a copy of Assignment B2. This does not solve the given problem.
44	18CS60R46	Sagun Tudu	1	0	0	0	**1**	A mindless copy of the solution of Assignment B2. This has nothing to do with the given problem.
45	18CS60R47	Shalini Saini	1	0	0	0	**1**	Why queue, heap, BST? This exercise has very little, if any, to do with Assignment B2.
46	18CS60R50	Rishabh Malhotra	1	9	4	2	**16**	Seg Fault. No need to copy to IL, IR, PL, PR. Use of totalnodes() along with populatesizes() is inefficient Only preorder printing done.

47	18CS60R51	Navdeep Khare	0	3	0	0	**3**	Nothing is implemented except for the search in inorder array.
48	18CS60R53	Somnath Hazra	1	8	0	2	**11**	createtree/buildtree conceptually OK, but has severa; bugs. Rest not implemented.
49	18CS60R54	Pratik Rawat	1	0	0	0	**1**	Mindless copy from Assignment B3. We are not dealing with BST's here.
50	18CS60R55	Aabhas Behere	0	2	0	0	**2**	Nothing appropriate is implemented. Even scanf()'s do not have &.
51	18CS60R56	Narayan Kunal	1	8	0	0	**9**	Complicated and faulty file input. BuildTree() logically OK, but has bugs. It should return a pointer. Don't pass the pointer. No need to copy arrays. Rest not done.
52	18CS60R57	Samriddhi Sanadhya	1	6	0	3	**10**	Bad C syntax and indentation. File input is bad. BuildTree() has bugs but conceptually not bad (but why n1 and n2). PrintPC should print subtree sizes.
53	18CS60R58	Shelke Yogesh Kalyan	2	9	0	2	**13**	Seg Fault. buildTree() algorithm fine. No effort of reading file. Sizes not handled.
54	18CS60R59	Deepak Kumar	5	9	4	5	**23**	No need to copy from I,P to IL,PL and IR,PR. You are not asked to store left and right subtree sizes. OK otherwise.
55	18CS60R60	Raj Kumar	0	0	0	0	**0**	No file submitted.
56	18CS60R61	Krishna Reddy Kopparthy	0	2	0	0	**2**	Nothing appropriate is implemented. Even node definition is incorrect. Indentation is very poor.
57	18CS60R62	Niraj Kumar Kachhwah	0	1	0	2	**3**	Code contains some (faulty) BST routines. The given problem is widely different.
58	18CS60R63	Nazmul Hussain	0	1	0	0	**1**	Only input, that too not from file.

59	18CS60R64	Mahendra Singh Kanyal	0	6	0	0	**6**	Seg Fault. BuildTree() is partially correct. Subarrays not needed for recursive calls. The logic behind copying to the subarrays is not clear. Rest not done. No file input.
60	18CS60R65	Rachit Agarwal	5	8	4	4	**21**	No file input. Copying arrays for recursive calls in buildtree() not needed. Size has nothing to do with data != 0. Otherwise, OK.
61	18CS60R66	Anjali Hotwani	0	2	0	0	**2**	Only the file is read (in a complicated way).
62	18CS60R67	Rishabh Waman Shahare	0	0	0	0	**0**	No file submitted.
63	18CS60R69	Himanshu Agarwal	0	0	0	2	**2**	Practically nothing is done.
64	18CS60R70	Telang Onkar Ajay	0	1	0	0	**1**	What is the program doing? No effort to solve the given problem. No file input.
65	18CS60R71	Sanjay Moharana	0	3	0	0	**3**	BuildTree() does not work as written in main(). No file input.
66	18CS60R72	Nishant Kumar Mishra	0	3	0	0	**3**	BuildTree() does not work as written in main(). No file input.
67	18CS60R73	Arun Jose	1	6	0	0	**7**	Program is full of bugs. Even 1-D array handling is faulty. ConstructTree() deserves some merits.
68	18CS60R75	Ritu Patel	0	0	0	2	**2**	Practically nothing is done. No file input. Indentation is horrible.
69	18CS60R76	Gunjan Balde	1	7	3	3	**14**	The program is full of bugs. In constructTreeUtil(), no need to copy for recursive calls. No input taken. Populatesizes() does not set node→size in all cases.
70	18CS60S01	Sudipta Paria	3	10	3	2	**18**	populatesizes() incomplete. PrintPC(): Only root→size is printed.

PYTHON PROGRAM (EVALUATED BY SPG)								
#	Roll No	Name	Output (5)	Read File (5)	Sort (10)	Print (5)	Total (25)	Comments
1	18CS60D01	Abdul Shamnar P	5	5	10	5	**25**	Perfect
2	18CS60D02	Amit Kumar Jha	2	3	0	2	**7**	Input not parsed; no datetime sort, sorted strings; printed in list format
3	18CS60D03	Bachau Prasad	0	2	4	0	**6**	Several syntactical errors; Sort function is not complete
4	18CS60D04	Ajay Kumar Gupta	5	5	10	3	**23**	Output Format not maintained
5	18CS60D05	Aditya Kaushik	0	3	9	3	**15**	input not properly parsed; output format not mainted
6	18CS60D06	Anish Poonia	5	5	10	5	**25**	Perfect
7	18CS60R01	Soham Poddar	5	5	10	5	**25**	Perfect
8	18CS60R02	Shounak Paul	5	5	10	5	**25**	Perfect
9	18CS60R03	Neha	5	5	10	3	**23**	Output format not maintained
10	18CS60R05	Himanshu Verma	3	5	8	3	**19**	month based sorting not properly done; output format not maintained

11	18CS60R07	Souradip Guha	5	5	10	5	**25**	Perfect
12	18CS60R08	Animesh Panwar	0	3	0	3	**6**	Input not parsed; no datetime specific sort; printed in list format
13	18CS60R10	Manish Kumar	0	5	0	0	**5**	only input parsing done in code
14	18CS60R11	Ahwan Mishra	5	5	10	5	**25**	Perfect
15	18CS60R12	Sugandh Pargal	2	5	5	3	**15**	not sorted; output format not maintained
16	18CS60R14	Sharad Shourya Ghosh	0	3	0	0	**3**	no output
17	18CS60R15	Gourav Chaturvedi	2	5	5	5	**17**	sorted w.r.t year only
18	18CS60R16	Erudakar Omkar Vasant	5	5	10	3	**23**	Output Format not maintained
19	18CS60R17	Udit Agrawal	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
20	18CS60R18	Shivangi Sharma	2	5	5	3	**15**	sorted w.r.t year only
21	18CS60R19	Rajesh Sahu	5	5	10	5	**25**	Perfect
22	18CS60R20	Abhijeet Bhandari	2	5	5	5	**17**	sorted w.r.t year only

23	18CS60R21	Subham Saha	5	5	10	5	**25**	Perfect
24	18CS60R22	Akashdeep	2	5	3	0	**10**	wrong algorithm; infinite loop; output format not maintained
25	18CS60R23	Birudaraju Sri Charan	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
26	18CS60R24	Aditi Singh	5	5	10	5	**25**	Perfect
27	18CS60R25	Nidamanuri Dharma Teja	5	5	10	5	**25**	Perfect
28	18CS60R26	Ayush Malik	2	5	8	5	**20**	sort logic not correct
29	18CS60R28	Shivansh Gupta	5	5	10	5	**25**	Perfect
30	18CS60R30	Karwa Prachi Mukesh	3	5	5	5	**18**	sorted w.r.t year only
31	18CS60R31	Bansi Shah	5	5	10	5	**25**	Perfect
32	18CS60R32	Subrata Chattopadhyay	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
33	18CS60R34	Vaibhav Mishra	2	5	10	3	**20**	output format not maintained
34	18CS60R35	Vishal Kumar	0	0	0	0	**0**	no code submitted

35	18CS60R36	Mayank Jain	2	5	10	4	**21**	some extra outputs in between;output format not maintained
36	18CS60R37	Surabhi S Kadur	2	3	0	2	**7**	Input not parsed; no sorting done; printed in list format
37	18CS60R38	Doshi Rushi Kamaleshbhai	5	5	10	5	**25**	Perfect
38	18CS60R39	Stanchion Bishoyi	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
39	18CS60R40	Harshita Chouhan	0	5	0	2	**7**	syntactical errors; no sorting done
40	18CS60R41	Saket Kumar	5	5	10	5	**25**	Perfect
41	18CS60R42	Gaurav Gupta	2	5	7	3	**17**	sort logic is wrong
42	18CS60R43	Pranjal Doshi	2	5	5	5	**17**	sorted w.r.t year only
43	18CS60R44	Amit Kumar	5	5	10	5	**25**	Perfect
44	18CS60R46	Sagun Tudu	2	0	0	2	**4**	syntactical errors; incomplete code
45	18CS60R47	Shalini Saini	2	3	2	2	**9**	syntactical errors; wrong logic
46	18CS60R50	Rishabh Malhotra	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained

47	18CS60R51	Navdeep Khare	2	5	5	5	**17**	dd, mm, yyyy sorted independently
48	18CS60R53	Somnath Hazra	0	2	0	0	**2**	only read the file
49	18CS60R54	Pratik Rawat	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
50	18CS60R55	Aabhas Behere	5	5	10	5	**25**	Perfect
51	18CS60R56	Narayan Kunal	5	5	10	5	**25**	Perfect
52	18CS60R57	Samriddhi Sanadhya	5	5	5	5	**20**	Works ok, 5 marks deducted for using external module lexsort
53	18CS60R58	Shelke Yogesh Kalyan	5	5	10	5	**25**	Perfect
54	18CS60R59	Deepak Kumar	0	0	0	0	**0**	Nothing is done
55	18CS60R60	Raj Kumar	2	5	6	4	**17**	sorted w.r.t year only; output format not maintained
56	18CS60R61	Krishna Reddy Kopparthy	2	5	8	4	**19**	syntactical errors; not converted month to int
57	18CS60R62	Niraj Kumar Kachhwah	2	3	6	2	**13**	list index out of range; sorted w.r.t. year only;output format not maintained
58	18CS60R63	Nazmul Hussain	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained

59	18CS60R64	Mahendra Singh Kanyal	0	0	0	0	**0**	no code submitted
60	18CS60R65	Rachit Agarwal	5	5	10	5	**25**	Perfect
61	18CS60R66	Anjali Hotwani	2	5	5	3	**15**	sorted w.r.t year only; output format not maintained
62	18CS60R67	Rishabh Waman Shahare	2	5	5	1	**13**	only sorted month number
63	18CS60R69	Himanshu Agarwal	1	5	3	2	**11**	wrong logic; runtime error;output format not maintained
64	18CS60R70	Telang Onkar Ajay	5	5	10	5	**25**	perfect
65	18CS60R71	Sanjay Moharana	5	5	10	5	**25**	perfect
66	18CS60R72	Nishant Kumar Mishra	5	5	10	5	**25**	perfect
67	18CS60R73	Arun Jose	5	5	10	5	**25**	perfect
68	18CS60R75	Ritu Patel	2	5	5	2	**14**	sorted w.r.t year only; output format not maintained
69	18CS60R76	Gunjan Balde	5	5	7	5	**22**	did not sort by date
70	18CS60S01	Sudipta Paria	0	5	0	2	**7**	no sorting done; only year is outputted

SUMMARY					
#	Roll No	Name	C (25)	PYTHON (25)	Total (50)
1	18CS60D01	Abdul Shamnar P	**8**	**25**	**33**
2	18CS60D02	Amit Kumar Jha	**7**	**7**	**14**
3	18CS60D03	Bachau Prasad	**7**	**6**	**13**
4	18CS60D04	Ajay Kumar Gupta	**3**	**23**	**26**
5	18CS60D05	Aditya Kaushik	**2**	**15**	**17**
6	18CS60D06	Anish Poonia	**7**	**25**	**32**
7	18CS60R01	Soham Poddar	**25**	**25**	**50**
8	18CS60R02	Shounak Paul	**24**	**25**	**49**
9	18CS60R03	Neha	**11**	**23**	**34**
10	18CS60R05	Himanshu Verma	**4**	**19**	**23**

11	18CS60R07	Souradip Guha	**12**	**25**	**37**
12	18CS60R08	Animesh Panwar	**7**	**6**	**13**
13	18CS60R10	Manish Kumar	**2**	**5**	**7**
14	18CS60R11	Ahwan Mishra	**5**	**25**	**30**
15	18CS60R12	Sugandh Pargal	**12**	**15**	**27**
16	18CS60R14	Sharad Shourya Ghosh	**2**	**3**	**5**
17	18CS60R15	Gourav Chaturvedi	**23**	**17**	**40**
18	18CS60R16	Erudakar Omkar Vasant	**17**	**23**	**40**
19	18CS60R17	Udit Agrawal	**7**	**15**	**22**
20	18CS60R18	Shivangi Sharma	**24**	**15**	**39**
21	18CS60R19	Rajesh Sahu	**2**	**25**	**27**
22	18CS60R20	Abhijeet Bhandari	**4**	**17**	**21**

23	18CS60R21	Subham Saha	**0**	**25**	**25**
24	18CS60R22	Akashdeep	**4**	**10**	**14**
25	18CS60R23	Birudaraju Sri Charan	**8**	**15**	**23**
26	18CS60R24	Aditi Singh	**18**	**25**	**43**
27	18CS60R25	Nidamanuri Dharma Teja	**12**	**25**	**37**
28	18CS60R26	Ayush Malik	**4**	**20**	**24**
29	18CS60R28	Shivansh Gupta	**19**	**25**	**44**
30	18CS60R30	Karwa Prachi Mukesh	**17**	**18**	**35**
31	18CS60R31	Bansi Shah	**16**	**25**	**41**
32	18CS60R32	Subrata Chattopadhyay	**14**	**15**	**29**
33	18CS60R34	Vaibhav Mishra	**4**	**20**	**24**
34	18CS60R35	Vishal Kumar	**4**	**0**	**4**

35	18CS60R36	Mayank Jain	**3**	**21**	**24**
36	18CS60R37	Surabhi S Kadur	**2**	**7**	**9**
37	18CS60R38	Doshi Rushi Kamaleshbhai	**4**	**25**	**29**
38	18CS60R39	Stanchion Bishoyi	**0**	**15**	**15**
39	18CS60R40	Harshita Chouhan	**2**	**7**	**9**
40	18CS60R41	Saket Kumar	**7**	**25**	**32**
41	18CS60R42	Gaurav Gupta	**2**	**17**	**19**
42	18CS60R43	Pranjal Doshi	**14**	**17**	**31**
43	18CS60R44	Amit Kumar	**1**	**25**	**26**
44	18CS60R46	Sagun Tudu	**1**	**4**	**5**
45	18CS60R47	Shalini Saini	**1**	**9**	**10**
46	18CS60R50	Rishabh Malhotra	**16**	**15**	**31**

47	18CS60R51	Navdeep Khare	**3**	**17**	**20**
48	18CS60R53	Somnath Hazra	**11**	**2**	**13**
49	18CS60R54	Pratik Rawat	**1**	**15**	**16**
50	18CS60R55	Aabhas Behere	**2**	**25**	**27**
51	18CS60R56	Narayan Kunal	**9**	**25**	**34**
52	18CS60R57	Samriddhi Sanadhya	**10**	**20**	**30**
53	18CS60R58	Shelke Yogesh Kalyan	**13**	**25**	**38**
54	18CS60R59	Deepak Kumar	**23**	**0**	**23**
55	18CS60R60	Raj Kumar	**0**	**17**	**17**
56	18CS60R61	Krishna Reddy Kopparthy	**2**	**19**	**21**
57	18CS60R62	Niraj Kumar Kachhwah	**3**	**13**	**16**
58	18CS60R63	Nazmul Hussain	**1**	**15**	**16**

59	18CS60R64	Mahendra Singh Kanyal	**6**	**0**	**6**
60	18CS60R65	Rachit Agarwal	**21**	**25**	**46**
61	18CS60R66	Anjali Hotwani	**2**	**15**	**17**
62	18CS60R67	Rishabh Waman Shahare	**0**	**13**	**13**
63	18CS60R69	Himanshu Agarwal	**2**	**11**	**13**
64	18CS60R70	Telang Onkar Ajay	**1**	**25**	**26**
65	18CS60R71	Sanjay Moharana	**3**	**25**	**28**
66	18CS60R72	Nishant Kumar Mishra	**3**	**25**	**28**
67	18CS60R73	Arun Jose	**7**	**25**	**32**
68	18CS60R75	Ritu Patel	**2**	**14**	**16**
69	18CS60R76	Gunjan Balde	**14**	**22**	**36**
70	18CS60S01	Sudipta Paria	**18**	**7**	**25**