

① previous one

Result :- C program of different datatypes was run successfully.

Output :- Enter details of 5 students:

Student 1:-

Roll no:- 1
Name:- Namita
Salary:- 9000
Marks:- 94

Phone no:- 9073339094

Student 3:-

Roll no:- 3
Name:- Rahul
Salary:- 8000
Marks:- 84

student 5:-

Roll no:- 5
Name:- P.
Salary:- 6000
Marks:- 60
Phone no:- 9999999999

Student 2:-

Roll no:- 2
Name:- Ishita
Salary:- 90,000
Marks:- 86

Phone no:- 9731959498

Student 4:-

Roll no:- 4
Name:- Shylika
Salary:- 80,000
Marks:- 96

Phone no:- 9353892005

② Code:-

Result :- C program to create atleast 5 members of different datatypes and user input was accepted

Output :- Enter details per person (id name age salary phone):

6. 1 namita 18 60,000 8073339094

7. 2 rahul 21 70,000 632845121

8. 3 shylika 23 50,000 807639495

9. 4 shanti 19 20,00 3216810015

10. 5 Priti 16 3000 8073339088

Result :- A student structure with at least 5 members was successfully created and displayed it.

Output

Enter roll no: 1

Enter name: dynamite

Enter age: 23

Enter grade(A-F): A

Enter address: pure

OP^v

Ans:

30

Output:-

Enter student details: ~~Rakhee~~

Enter name: ~~Rakhee~~

Enter roll no: ~~677~~

Enter marks: ~~94~~

Enter phone number: ~~8073339094~~

Enter street: ~~FC road~~

Enter city: ~~Pune~~

Enter state: ~~Maharashtra~~

Enter zipcode: ~~410302~~

Enter (Dob): ~~30~~

Enter (mm): ~~06~~

Enter (yyyy): - ~~2005~~

Display all

Result:- { program for nested
structures for members D.O.B.
and address was

Successfully run.

(W)

Output :-

Enter the number of rows for matrix 1: 2
Enter the number of columns for matrix 1: 2
Enter the number of rows for matrix 2: 2
Enter the number of columns for matrix 2: 2

Enter the elements of matrix 1:

enter the elements [1][1]: 1 ✓

enter the elements [1][2]: 1 ✓

enter the elements [2][1]: 1 ✓

enter the elements [2][2]: 1 ✓

Enter the elements of matrix 2:

enter the elements [1][1]: 1 ✓

enter the elements [1][2]: 1 ✓

enter the elements [2][1]: 1 ✓

enter the elements [2][2]: 1 ✓

Resultant matrix:

2 2
2 2

O.P.Q

Result :-

Matrix Multiplication in C was
successfully run.

Output:-

1. Insert element
2. Delete element
3. Print list
4. Exit

~~choice~~ your choice : 1

Enter your choice to insert : 2 3 4 5 6

Enter elements to insert : 1 2 3 4 5 6

Enter your choice : 3

Elements are :- 2, 3, 4, 5, 6

Enter your choice : 4

Exit

SDP
Date 13/8/24

Result :- The code was in C language
was successfully run.

Output :-

Menu:

1. Insert:
2. delete:
3. Search:
4. display:
5. Exit:

enter your choice :- 1.

enter the element to insert : 5

5 inserted successfully.

Menu:

1. Insert
2. delete
3. Search
4. display
5. Exit

enter your choice : 3

enter the element to search : 5

target 5 found.

MENU:

1. Insert:
2. Delete:
3. Search:
4. Display:
5. Exit:

enter your choice : 5

Mr. Shekhar
22/8/2024

exiting program.

Output :-

Menu:

1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Delete first Node
5. Delete last Node
6. Delete node at position
7. Search Node
8. Display List
9. Exit

Enter your choice : 2

Enter data to insert : 456

Enter data to insert : 123

Enter data to insert : 3

Enter data to insert : 789

Enter position : 1

Enter your choice : 3

123 → 789 → 456 → NULL.

S. Mathur
22.08.24

Output:-

1. Insert at Beginning
2. Insert at End
3. Insert at Position
4. Delete first node.
5. Delete last node.
6. Delete node at position.
7. Search Node
8. Display List

Enter your choice : 1

Enter data to insert at beginning : 10

Enter your choice : 9

exit(0).

Result :- The Doubly linked list in C code was run successfully.

Output:-

Stack operations:

1. Push
2. Pop
3. Display
4. Exit

Enter your choice: 1

Enter value to push: 7

Pushed 7 onto the stack.

Stack operations:

1. Push
2. Pop
3. Display
4. Exit

Enter your choice : 5

exit(0);

Result:- The code was run successfully using stack using

Array.

Output:-

Stack operations:

1) Push

2) Pop

3) Display

4) Exit

Enter your choice: 1

Enter value to push: 8

Pushed 8 onto the stack.

stack = 8 ← stack

Output:-

Menu :

1. Enqueue
2. Dequeue
3. Transverse
4. Exit

Enter your choice: 2
Queue is empty!

- Menu:
1. Enqueue
 2. Dequeue
 3. Transverse
 4. Exit

Enter your choice: ④

exit

exit
exit

Output:-

1. Convert Infix to Postfix

2. exit

Enter your choice : 1

Enter ~~expr~~ the infix expression : a * b * c

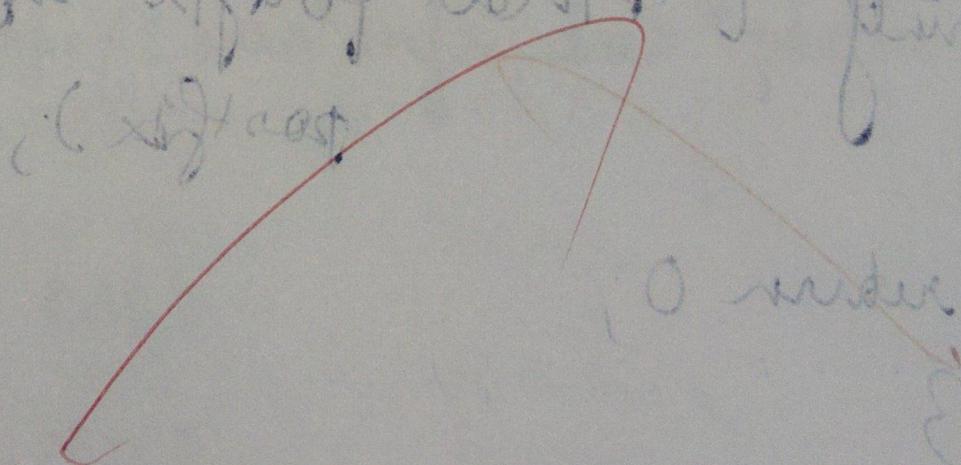
Reading character : *

Postfix expression : a

Reading character : +

Postfix expression :

Stacking tokens Kifbag level 1



Output :- Enter number of elements in the tree :
Enter 2 integers : 48 and 909
3 trees to choose from.

48

909: x AM Tree

y AM Tree

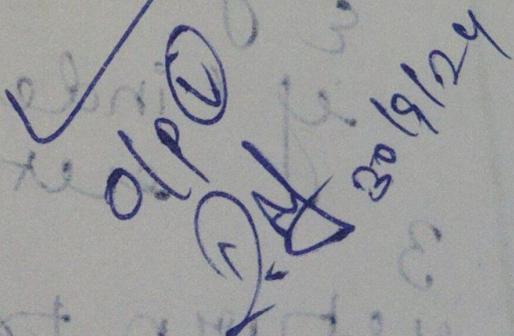
z AM Tree

Menu:

1. Find Parent
2. In-order Transversal
3. Pre-order Transversal
4. Post-order Transversal
5. Search Node
6. Exit

Enter your choice: 2

In-order Transversal : 909 48



Output:-

Binary search tree menu

1. Insert a node.
2. Delete a node.
3. Perform inorder Traversal.
4. Exit

Enter your choice : 1

Enter the key to insert : 4

Binary search tree menu.

1. Insert a node.
2. Delete a node.
3. Perform inorder Traversal
4. Exit

Enter your choice : 4

Exit....

Code Executed successfully.

Result:- Code was successfully run
of implementing a binary
search tree.

Or?

- Output :-
1. Insertion.
 2. Deletion.
 3. Traversal.
 4. Exit.

Enter your choice : 1.

Enter the item to insert : 4, 6, 7, 8.

1. Insertion
2. Deletion
3. Traversal
4. Exit.

Invalid choice!

Result:- The code of C was run successfully.