CODE:::

```python
#!/usr/bin/env python
# coding: utf-8




tab = []
result = []
goalList = ["a", "b", "c", "d", "e"]




def parSolution(N):
    for i in range(N):
        if goalList[i] != result[i]:
            return False
    return True




def Onblock(index, count):

    # break point of recursive call
    if count == len(goalList)+1:
        return True
    # copy tab of index value to result
    block = tab[index]
    # stack block
    result.append(block)
    print(result)
    if parSolution(count):
        print("Pushed a result solution ")
        # remove block from tab
        tab.remove(block)
        Onblock(0, count + 1)
    else:
        print("result solution not possible, back to the tab")
        # pop out if no partial solution
        result.pop()
        Onblock(index+1, count)




def Ontab(problem):
    # check if everything in stack is on the tab
    if len(problem) != 0:
        tab.append(problem.pop())
        Ontab(problem)
    # if everything is on the tab the we return true
    else:
        return True




def goal_stack_planing(problem):
    # pop problem and put in tab
```

```python
    Ontab(problem)
    # print index and number of blocks on result stack
    if Onblock(0, 1):
        print(result)


if __name__ == "__main__":
    problem = ["c", "a", "e", "d", "b"]
    print("Goal Problem")
    for k, j in zip(goalList, problem):
        print(k+"    "+j)
    goal_stack_planing(problem)
    print("result Solution")
    print(result)
```
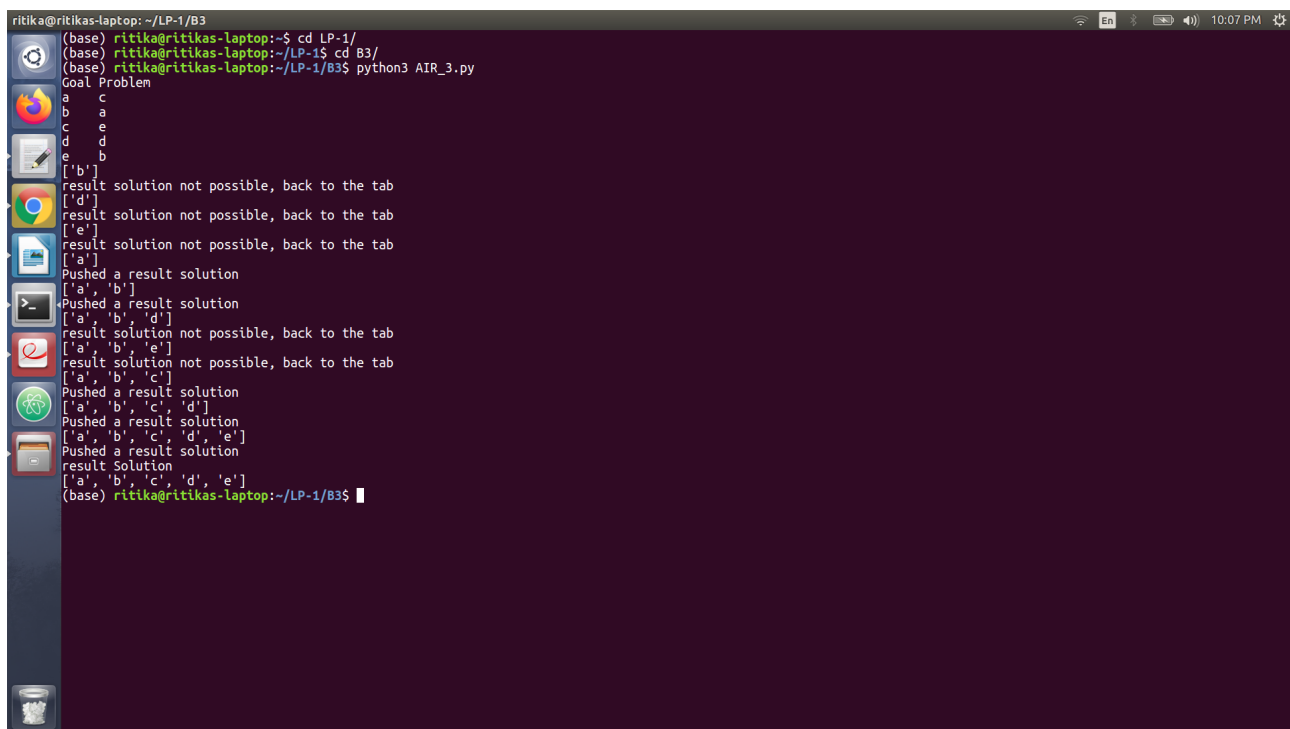
OUTPUT:::

___