1. **Programming/Scripting/Database Format**
3. **Question Type\*** : Single Question
4. **Question Subtype\*** : Single File Programming Questions
5. **Question Metadata**
6. Manual Difficulty\* : HIGH
7. Sub Topic\* :   Resource allocation
8. Topic\* :  Banker's Algorithm
9. Subject\* :  Operating Systems
10. **Question Details**
11. Question \* :

Write a program to show the working of Banker's Algorithm in Operating System (OS)

3. Language\* : Java
4. Input Format :  Hardcoded
5. Output Format :
6. Code Constraints :
7. **Solutions & Testcases**
8. Solution\* :
9. // import required classes and packages
10. **import** java.util.\*;
11. **import** java.io.\*;
12. **import** java.util.Scanner;
14. // create BankersAlgoExample class to implement Banker's algorithm in Java
15. **class** BankersAlgoExample
16. {
18. // create findNeedValue() method to calculate the need of each process
19. **static** **void** findNeedValue(**int** needArray[][], **int** maxArray[][], **int** allocationArray[][], **int** totalProcess, **int** totalResources)
20. {
21. // use nested for loop to calculate Need for each process
22. **for** (**int** i = 0 ; i < totalProcess ; i++){    // for each process
23. **for** (**int** j = 0 ; j < totalResources ; j++){  //for each resource
24. needArray[i][j] = maxArray[i][j] - allocationArray[i][j];
25. }
26. }
27. }
29. // create checkSafeSystem() method to determine whether the system is in safe state or not
30. **static** **boolean** checkSafeSystem(**int** processes[], **int** availableArray[], **int** maxArray[][], **int** allocationArray[][], **int** totalProcess, **int** totalResources)
31. {
32. **int** [][]needArray = **new** **int**[totalProcess][totalResources];
34. // call findNeedValue() method to calculate needArray
35. findNeedValue(needArray, maxArray, allocationArray, totalProcess, totalResources);
37. // all the process should be infinished in starting
38. **boolean** []finishProcesses = **new** **boolean**[totalProcess];
40. // initialize safeSequenceArray that store safe sequenced
41. **int** []safeSequenceArray = **new** **int**[totalProcess];
43. // initialize workArray as a copy of the available resources
44. **int** []workArray = **new** **int**[totalResources];
46. **for** (**int** i = 0; i < totalResources ; i++)    //use for loop to copy each available resource in the workArray
47. workArray[i] = availableArray[i];
49. // initialize counter variable whose value will be 0 when the system is not in the safe state or when all the processes are not finished.
50. **int** counter = 0;
52. // use loop to iterate the statements until all the processes are not finished
53. **while** (counter < totalProcess)
54. {
55. // find infinished process which needs can be satisfied with the current work resource.
56. **boolean** foundSafeSystem = **false**;
57. **for** (**int** m = 0; m < totalProcess; m++)
58. {
59. **if** (finishProcesses[m] == **false**)        // when process is not finished
60. {
61. **int** j;
63. //use for loop to check whether the need of each process for all the resources is less than the work
64. **for** (j = 0; j < totalResources; j++)
65. **if** (needArray[m][j] > workArray[j])      //check need of current resource for current process with work
66. **break**;
68. // the value of J and totalResources will be equal when all the needs of current process are satisfied
69. **if** (j == totalResources)
70. {
71. **for** (**int** k = 0 ; k < totalResources ; k++)
72. workArray[k] += allocationArray[m][k];
74. // add current process in the safeSequenceArray
75. safeSequenceArray[counter++] = m;
77. // make this process finished
78. finishProcesses[m] = **true**;
80. foundSafeSystem = **true**;
81. }
82. }
83. }
85. // the system will not be in the safe state when the value of the foundSafeSystem is false
86. **if** (foundSafeSystem == **false**)
87. {
88. System.out.print("The system is not in the safe state because lack of resources");
89. **return** **false**;
90. }
91. }
93. // print the safe sequence
94. System.out.print("The system is in safe sequence and the sequence is as follows: ");
95. **for** (**int** i = 0; i < totalProcess ; i++)
96. System.out.print("P"+safeSequenceArray[i] + " ");
98. **return** **true**;
99. }
101. // main() method start
102. **public** **static** **void** main(String[] args)
103. {
104. **int** numberOfProcesses, numberOfResources;
106. //create scanner class object to get input from user
107. Scanner sc = **new** Scanner(System.in);
109. // get total number of resources from the user
110. System.out.println("Enter total number of processes");
111. numberOfProcesses = sc.nextInt();
113. // get total number of resources from the user
114. System.out.println("Enter total number of resources");
115. numberOfResources = sc.nextInt();
117. **int** processes[] = **new** **int**[numberOfProcesses];
118. **for**(**int** i = 0; i < numberOfProcesses; i++){
119. processes[i] = i;
120. }
122. **int** availableArray[] = **new** **int**[numberOfResources];
123. **for**( **int** i = 0; i < numberOfResources; i++){
124. System.out.println("Enter the availability of resource"+ i +": ");
125. availableArray[i] = sc.nextInt();
126. }
128. **int** maxArray[][] = **new** **int**[numberOfProcesses][numberOfResources];
129. **for**( **int** i = 0; i < numberOfProcesses; i++){
130. **for**( **int** j = 0; j < numberOfResources; j++){
131. System.out.println("Enter the maximum resource"+ j +" that can be allocated to process"+ i +": ");
132. maxArray[i][j] = sc.nextInt();
133. }
134. }
136. **int** allocationArray[][] = **new** **int**[numberOfProcesses][numberOfResources];
137. **for**( **int** i = 0; i < numberOfProcesses; i++){
138. **for**( **int** j = 0; j < numberOfResources; j++){
139. System.out.println("How many instances of resource"+ j +" are allocated to process"+ i +"? ");
140. allocationArray[i][j] = sc.nextInt();
141. }
142. }
144. //call checkSafeSystem() method to check whether the system is in safe state or not
145. checkSafeSystem(processes, availableArray, maxArray, allocationArray, numberOfProcesses, numberOfResources);
146. }
147. }

**Test case: Input**

java -cp /tmp/x48hYQupWG BankersAlgoExample

Enter total number of processes

5

Enter total number of resources3

Enter the availability of resource0:

3

Enter the availability of resource1:

3

Enter the availability of resource2:

2

Enter the maximum resource0 that can be allocated to process0:

7

Enter the maximum resource1 that can be allocated to process0:

5

Enter the maximum resource2 that can be allocated to process0:

3

Enter the maximum resource0 that can be allocated to process1:

3

Enter the maximum resource1 that can be allocated to process1:

2

Enter the maximum resource2 that can be allocated to process1: 2

Enter the maximum resource0 that can be allocated to process2: 9

Enter the maximum resource1 that can be allocated to process2:

0

Enter the maximum resource2 that can be allocated to process2:

2

Enter the maximum resource0 that can be allocated to process3:

2

Enter the maximum resource1 that can be allocated to process3:

2

Enter the maximum resource2 that can be allocated to process3:

2

Enter the maximum resource0 that can be allocated to process4: 4

Enter the maximum resource1 that can be allocated to process4:

3

Enter the maximum resource2 that can be allocated to process4:

3

How many instances of resource0 are allocated to process0?

0

How many instances of resource1 are allocated to process0?

1

How many instances of resource2 are allocated to process0?

0

How many instances of resource0 are allocated to process1? 2

How many instances of resource1 are allocated to process1?

0

How many instances of resource2 are allocated to process1?

0

How many instances of resource0 are allocated to process2?

3

How many instances of resource1 are allocated to process2?

0

How many instances of resource2 are allocated to process2?

2

How many instances of resource0 are allocated to process3?

2

How many instances of resource1 are allocated to process3?

1

How many instances of resource2 are allocated to process3? 1

How many instances of resource0 are allocated to process4?

0

How many instances of resource1 are allocated to process4?

0

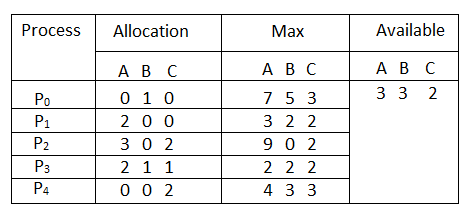
How many instances of resource2 are allocated to process4?

2

**Expected OutCome:**

The system is in safe sequence and the sequence is as follows: P1 P3 P4 P0 P2

**Hidden Test Case:**



**Expected output:**

java -cp /tmp/LwMaBrIVMb BankersAlgoExample

Enter total number of processes

5

Enter total number of resources3

Enter the availability of resource0:

3

Enter the availability of resource1:

3

Enter the availability of resource2: 2

Enter the maximum resource0 that can be allocated to process0: 7

Enter the maximum resource1 that can be allocated to process0:

5

Enter the maximum resource2 that can be allocated to process0: 3

Enter the maximum resource0 that can be allocated to process1:

3

Enter the maximum resource1 that can be allocated to process1:

2

Enter the maximum resource2 that can be allocated to process1:

2

Enter the maximum resource0 that can be allocated to process2:

9

Enter the maximum resource1 that can be allocated to process2: 0

Enter the maximum resource2 that can be allocated to process2:

2

Enter the maximum resource0 that can be allocated to process3:

2

Enter the maximum resource1 that can be allocated to process3:

2

Enter the maximum resource2 that can be allocated to process3:

2

Enter the maximum resource0 that can be allocated to process4:

4

Enter the maximum resource1 that can be allocated to process4:

3

Enter the maximum resource2 that can be allocated to process4:

3

How many instances of resource0 are allocated to process0?

0

How many instances of resource1 are allocated to process0?

1

How many instances of resource2 are allocated to process0?

0

How many instances of resource0 are allocated to process1?

2

How many instances of resource1 are allocated to process1?

0

How many instances of resource2 are allocated to process1? 0

How many instances of resource0 are allocated to process2?

3

How many instances of resource1 are allocated to process2?

0

How many instances of resource2 are allocated to process2?

2

How many instances of resource0 are allocated to process3?

2

How many instances of resource1 are allocated to process3?

1

How many instances of resource2 are allocated to process3?

1

How many instances of resource0 are allocated to process4? 0

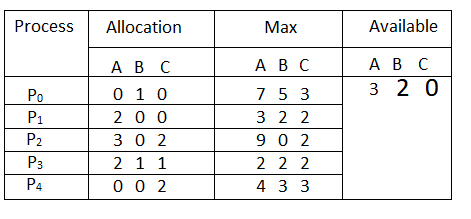
How many instances of resource1 are allocated to process4?

0

How many instances of resource2 are allocated to process4? 2

**The system is in safe sequence and the sequence is as follows: P1 P3 P4 P0 P2**

**Negative Test Case:**

****

Enter total number of processes 5

Enter total number of resources

3

Enter the availability of resource0:

3

Enter the availability of resource1: 2

Enter the availability of resource2:

0

Enter the maximum resource0 that can be allocated to process0:

7

Enter the maximum resource1 that can be allocated to process0:

5

Enter the maximum resource2 that can be allocated to process0:

3

Enter the maximum resource0 that can be allocated to process1:

3

Enter the maximum resource1 that can be allocated to process1:

2

Enter the maximum resource2 that can be allocated to process1:

2

Enter the maximum resource0 that can be allocated to process2:

9

Enter the maximum resource1 that can be allocated to process2:

0

Enter the maximum resource2 that can be allocated to process2:

2

Enter the maximum resource0 that can be allocated to process3:

2

Enter the maximum resource1 that can be allocated to process3:

2

Enter the maximum resource2 that can be allocated to process3:

2

Enter the maximum resource0 that can be allocated to process4:

4

Enter the maximum resource1 that can be allocated to process4:

3

Enter the maximum resource2 that can be allocated to process4:

3

How many instances of resource0 are allocated to process0?

0

How many instances of resource1 are allocated to process0?

1

How many instances of resource2 are allocated to process0?

0

How many instances of resource0 are allocated to process1?

2

How many instances of resource1 are allocated to process1?

0

How many instances of resource2 are allocated to process1?

0

How many instances of resource0 are allocated to process2?

3

How many instances of resource1 are allocated to process2?

0

How many instances of resource2 are allocated to process2?

2

How many instances of resource0 are allocated to process3?

2

How many instances of resource1 are allocated to process3?

1

How many instances of resource2 are allocated to process3?

1

How many instances of resource0 are allocated to process4?

0

How many instances of resource1 are allocated to process4? 0

How many instances of resource2 are allocated to process4?

2

**Expected output:** **The system is not in the safe state because lack of resources**

1. **Programming/Scripting/Database Format**
3. **Question Type\*** : Single Question
4. **Question Subtype\*** : Single File Programming Questions
5. **Question Metadata**
6. Manual Difficulty\* : Medium
7. Sub Topic\* :   Shell script
8. Topic\* : Prime numbers vis shell scripting
9. Subject\* :  Operating Systems
10. **Question Details**
11. Question \* :

Write shell script to determine whether a number is a prime or not

3. Language\* : Shell Scripting
4. Input Format :  Hardcoded
5. Output Format :
6. Code Constraints:
7. **Solutions & Testcases**
8. Solution\* :

**Online Compiler:** [**https://www.onlinegdb.com/online\_bash\_shell**](https://www.onlinegdb.com/online_bash_shell)

Directions: # Write your code in this editor and press "Run" button to execute it.

**Sample Solution**:

echo "Hello World";

#!/bin/bash

echo -e "Enter Number : \c"

read n

for((i=2; i<=$n/2; i++))

do

ans=$(( n%i ))

if [ $ans -eq 0 ]

then

echo "$n is not a prime number."

exit 0

fi

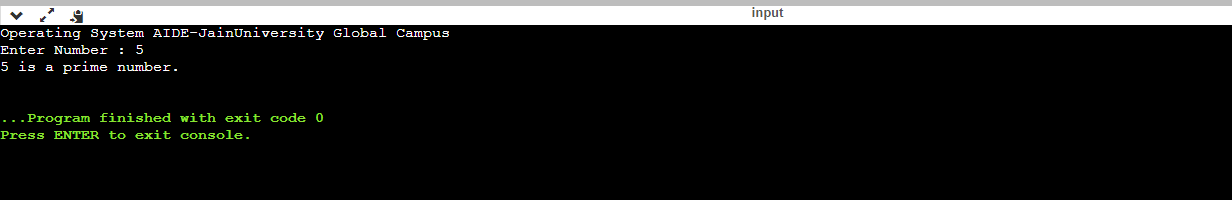
done

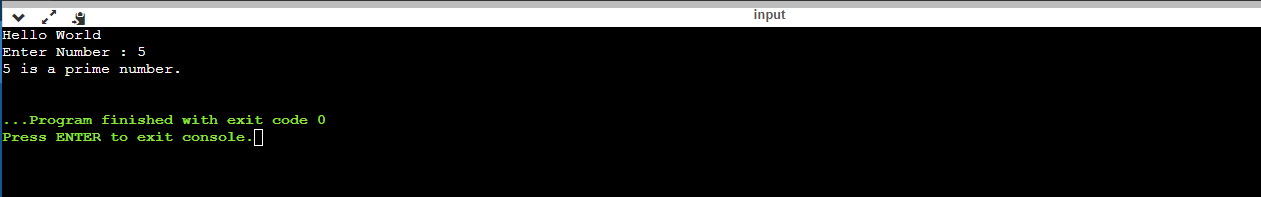
echo "$n is a prime number."

**Expected Output:**

Enter Number: 5

Output: 5 is a prime number.



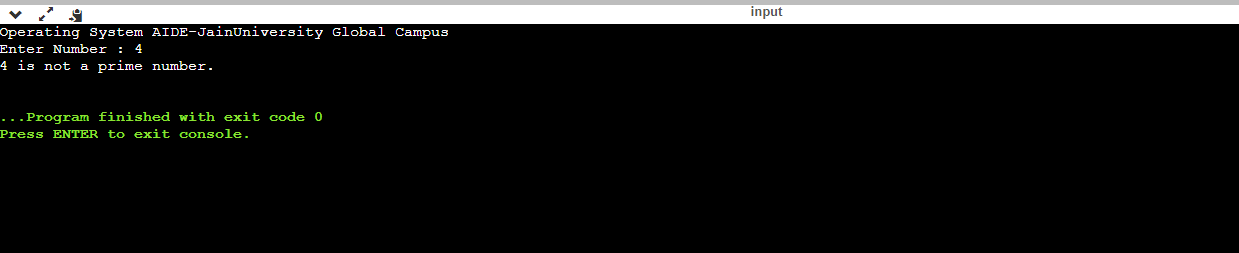


**Sample Test cases:**

Negative test case:

Enter input: 4

Expected output: 4 is not a prime number.



1. **Programming/Scripting/Database Format**
3. **Question Type\*** : Single Question
4. **Question Subtype\*** : Single File Programming Questions
5. **Question Metadata**
6. Manual Difficulty\* : Medium
7. Sub Topic\* :   Shell script
8. Topic\* :  Reverse of a number
9. Subject\* :  Operating Systems
10. **Question Details**
11. Question \* :

Write a shell script to find reverse of a Entered number

3. Language\* : Shell Scripting
4. Input Format :  Hardcoded
5. Output Format :
6. Code Constraints :
7. **Solutions & Testcases**
8. Solution\* :

**Sample solution:**

echo "Operating System AIDE-JainUniversity Global Campus";

#!/bin/bash

echo "Enter a Number:"

read a

rev=0

sd=0

or=$a

while [ $a -gt 0 ]

do

sd=`expr $a % 10`

temp=`expr $rev \\* 10`

rev=`expr $temp + $sd`

a=`expr $a / 10`

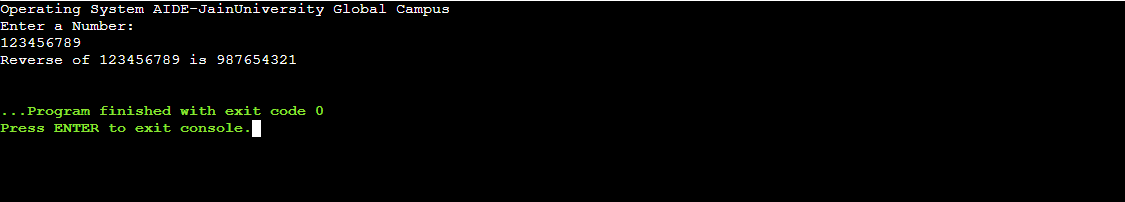
done

echo "Reverse of $or is $rev"

**Expected Output:**

Enter number: 123456789

Expected Output: 987654321



1. **Programming/Scripting/Database Format**
3. **Question Type\*** : Single Question
4. **Question Subtype\*** : Single File Programming Questions
5. **Question Metadata**
6. Manual Difficulty\* : Medium
7. Sub Topic\* :   Shell script
8. Topic\* :  palindrome
9. Subject\* :  Operating Systems
10. **Question Details**
11. Question \* :

Write a shell script to find a Entered number is a palindrome or not

3. Language\* : Shell Scripting
4. Input Format :  Hardcoded
5. Output Format :
6. Code Constraints :
7. **Solutions & Testcases**
8. Solution\* :

**Solution:**

echo "Operating System AIDE Jain University GLobal Campus";

echo "Enter the number"

read n

function pal

{

number=$n

reverse=0

while [ $n -gt 0 ]

do

a=`expr $n % 10 `

n=`expr $n / 10 `

reverse=`expr $reverse \\* 10 + $a`

done

echo $reverse

if [ $number -eq $reverse ]

then

echo "Number is palindrome"

else

echo "Number is not palindrome"

fi

}

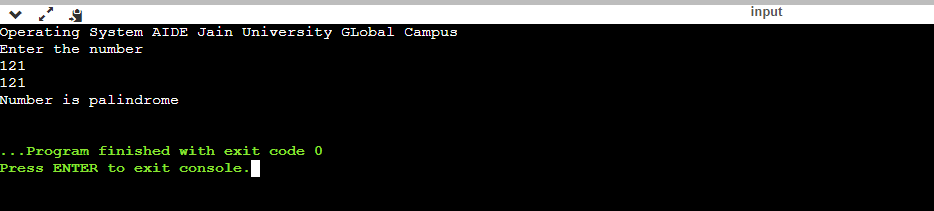
r=`pal $n`

echo "$r"

**Expected Output:**

**Enter number: 121**

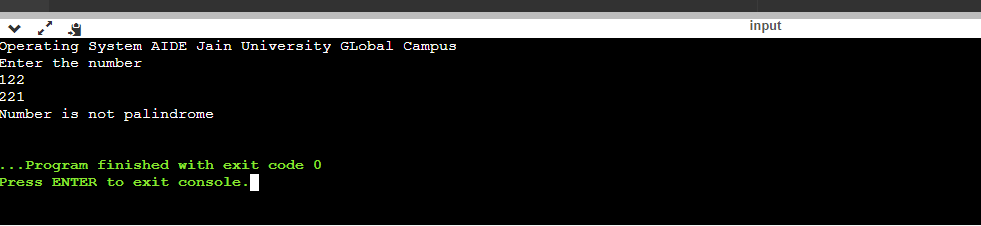
**Output: 121, Number is Palindrome.**

****

**Negative Test Case:**

**Enter Number: 122**

**Output:221, Number is not palindrome.**

****