1. Write a program to find out the distinct elements from given set of integer array.

**For example**: If given array elements are 23, 43, 34, 56, 87, 3

**Output should be**: 2 3 4 5 6 8 7

***Note:*** *There should not be any duplicacy in output elements.*

1. Return an array that contains exactly the same numbers as the given array, but rearranged so that every 3 is immediately followed by 4 & don’t move position of the 3's.

(Consider that fact that in the given array 3 can’t be adjacent to another 3 i.e. 3 followed by another 3)

**For example:**

rearrange([1, 3, 1, 1]) → [1, 3, 1, 1]

rearrange([1, 3, 1, 4]) → [1, 3, 4, 1]

rearrange([1, 3, 1, 4, 3, 1]) → [1, 3, 4, 1, 3, 1]

rearrange([1, 3, 1, 4, 3, 1, 4]) → [1, 3, 4, 1, 3, 4, 1] (Position of 3 is fixed while position of other elements can be different)

1. Write a program to check whether the group of N numbers at the start and end of the given array are same OR not (Take input for array & number from the console)

**For example:**

checkEnds([5, 6, 45, 99, 13, 5, 6], 1) → false

checkEnds ([5, 6, 45, 99, 13, 5, 6], 2) → true

checkEnds ([5, 6, 45, 99, 13, 5, 6], 3) → false

1. Write a program to print the following pattern upto nth row. Starting from 3, number is sum of previous two numbers like 3 is sum of 1 & 2, **8** is sum of 3 & 5, 13 is sum of 5 & 8 and so on.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 | 3 |  |  |  |
| 5 | 8 | 13 |  |  |
| 21 | 34 | 55 | 89 |  |
| 144 | 233 | 377 | 610 | 987 |

1. Consider the following table and columns described below:

**Project Details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project\_ID | Project\_Name | Manager | Start\_Date | End\_Date |

5 (a) Write a query to list all the projects which are having name ending with “D”

5(b) Write a query to find the name of the managers who are working on more than 1 project.

5(c) Write a query to count the number of projects whose start date & end date is between 2016-06-20 and 2016-06-19.

1. Consider the database structure as shown below:

**Table: Order Table: Product**

|  |  |
| --- | --- |
| Order\_ID | Order\_Date |
| 0001 | 2016-06-24 |
| 0002 | 2016-06-26 |
| 0003 | 2016-06-28 |
| 0004 | 2016-06-30 |

|  |  |  |
| --- | --- | --- |
| P\_ID | P\_NAME | Status |
| 0001 | Nokia 460 | Active |
| 0002 | MI Note 3 | Inactive |
| 0003 | Samsung J2 | Active |
| 0004 | Lenovo Note 4 | Active |

**Table: Order\_Product**

|  |  |  |
| --- | --- | --- |
| ORDER\_ID | P\_ID | Quantity |
| 0001 | 0001 | 1 |
| 0001 | 0002 | 1 |
| 0002 | 0004 | 4 |
| 0003 | 0002 | 1 |

6(a) Write down possible Index, Keys (Primary key, foreign keys etc), Data type, size of each columns.

6(b) Write a query to find out the order Id & order date of Nokia 460.

6(c) Write a query to find out the order number(s) and Order Date(s) having more than 1 product.

6(d) Write a query to find the name of the products & their corresponding order count. Output should be:

**Name OrderCount**

Nokia 460 1

MI Note 3 2

Samsung J2 0

Lenovo Note 4 1