Assignment 4

Introduction to programming in C

Question 1

Given two arrays of positive integers, output the largest number in the first array not present in the second one.

Input

The first line contains the size of the first array. The second line contains the contents of first array. The third line contains the size of the second array. The fourth line contains the contents of second array.

Output Output the largest number occurring in the first array that does not occur in the second. In case there is no such number, output 0.

Note : The sizes of the arrays are smaller than 20.

Solution

```
#include<stdio.h>

#define MAX 20

int read_array(int arr[]) {
  int i,n;
  scanf(”%d”,&n);
  for (i = 0; i < n; i++)
    scanf(”%d”, & arr[i]);
  return n;
}

int present(int arr[], int n, int elt) {
  int i;
  for (i = 0; i < n; i++) {
    if (arr[i] == elt) {
      return 1;
    }
  }
  return 0;
}


```

```c
int main() {
  int n1,n2;
  int arr1[MAX];
  int arr2[MAX];

  n1 = read_array(arr1);
  n2 = read_array(arr2);

  int i, largest = 0;

  for (i = 0; i < n1; i++) {
    int val = arr1[i];
    if (!present(arr2, n2, val)) {
      if (largest < val) {
        largest = val;
      }
    }
  }

  printf("%d", largest);

  return 0;
}
```

Question 2

Given a sequence of integers, find the number of distinct numbers in the sequence. The sequence need not be sorted.

Input

The input consists of two lines. The first line consists of a positive number N (N is at most 1000). The second line consists of N integers separated by spaces.

Output

The number of distinct elements in the sequence.

Solution

```c
#include <stdio.h>

#define SIZE 1000
int main() {
  int arr[SIZE];
  int i,j,n;
  int count = 0;

  scanf("%d", &n);

  for (i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
  }

  for (i = 0; i < n; i++) {
    int found = 0;/*see if arr[i] occurs previously in the array */
    for (j = i - 1; j >= 0; j--) {
      if (arr[j] == arr[i]) {
        found = 1;
        break;
      }
    }
    if (found == 0) {
      /* arr[i] is a new element */
      count = count + 1;
    }
  }

  printf("%d\n", count);
  return 0;
}

int main() {
  int k;
  scanf("%d", & k);
  find_odd(k);
  return 0;
}
```

Question 3

Write a program that replaces the occurrence of a given character (say c) in a primary string (say PS) with another string (say s).

Input

The first line contains the primary string (PS) The next line contains a character (c) The next line contains a string (s)

Output

Print the string PS with every occurence of c replaced by s.

NOTE:- There are no whitespaces in PS or s. Maximum length of PS is 100. Maximum length of s is 10.

Solution

```
#include<stdio.h>

int length(char * s) {
  int i = 0;
  while (s[i] != '\0') {
    i++;
  }
  return i;
}

```

```c
void Replace(char * str, char * sub, int start, int end) {
  int i = -1;
  int len = length(str);
  char temp[100];
  do {
    i++;
    temp[i] = str[i];
  } while (i != len);
  int sublen = length(sub);
  int j = start;
  for (i = 0; i < sublen; i++) {
    str[j++] = sub[i];
  }
  for (i = end + 1; i <= len; i++) {
    str[j++] = temp[i];
  }
}

int charCheck(char * str, char a, int i) {
  if (str[i] == a)
    return i;
  else
    return -1;
}

int main() {
  char str[100] = {0};
  int i = -1, start = 0, end = 0;
  scanf("%s", str);
  char c1;
  scanf("\n%c", & c1);
  char subs1[15];
  scanf("%s", subs1);
  for (i = 0; i < length(str); i++) {
    int val = charCheck(str, c1, i);
    if (val >= 0) {
      start = i;
      end = val;
      Replace(str, subs1, start, end);
      i = start + length(subs1) - 1;
    }
  }
  printf("%s", str);
  return 0;
}
```