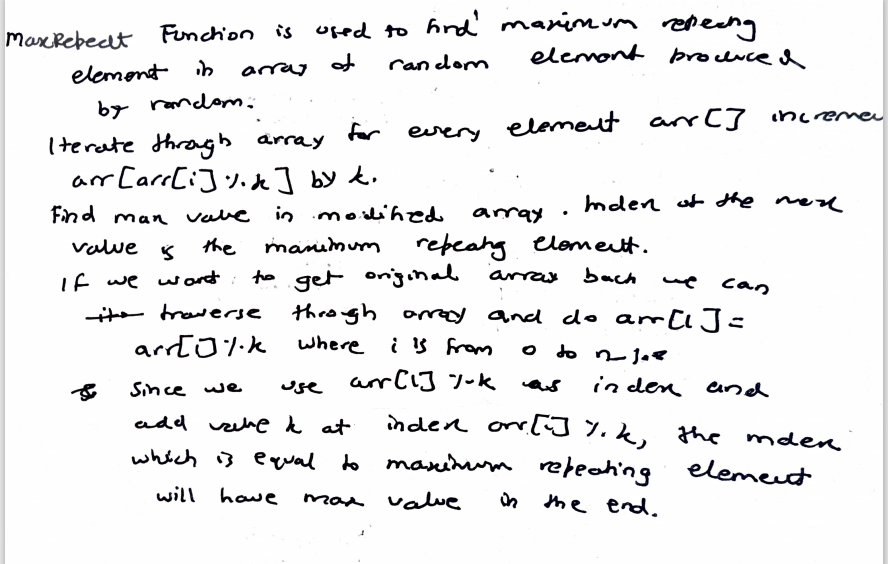
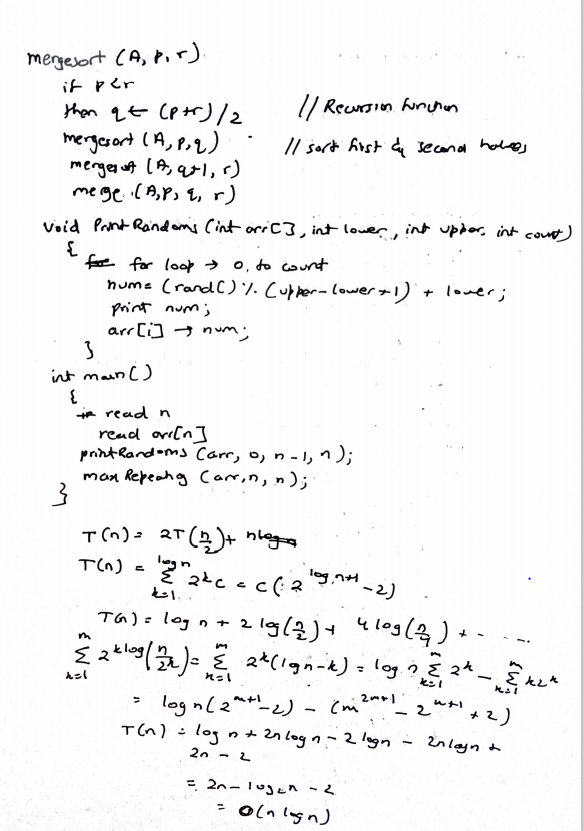
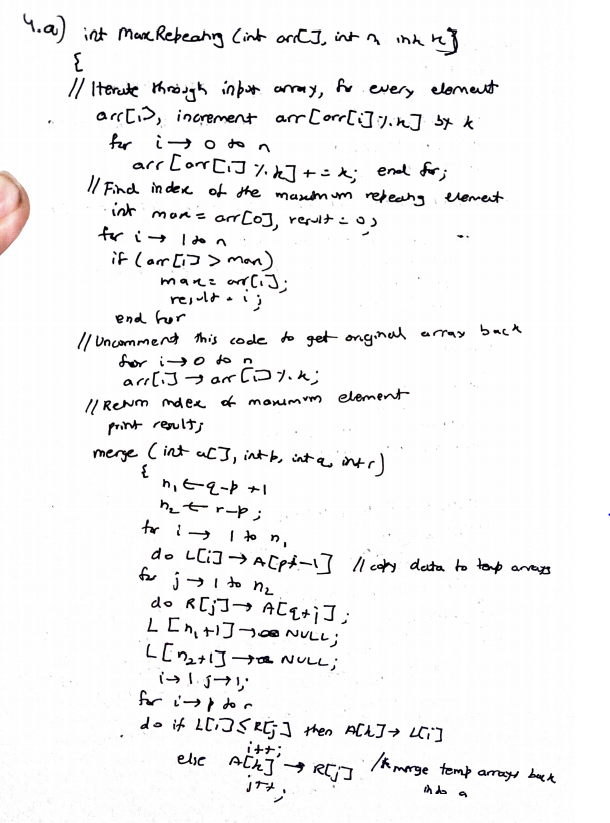
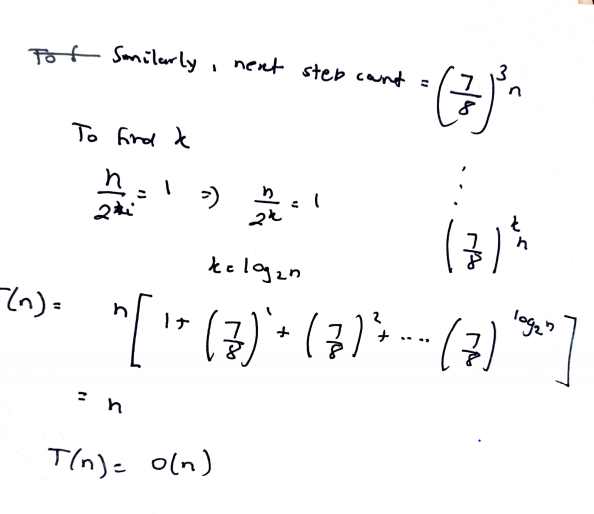
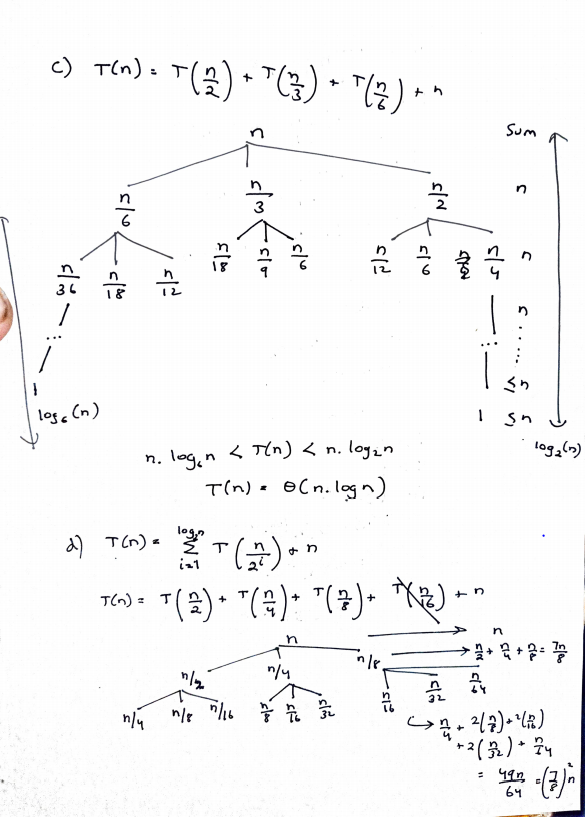
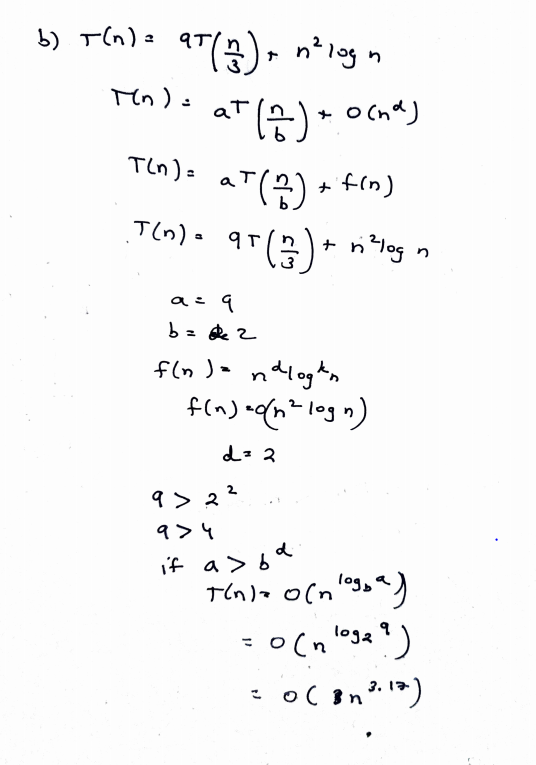
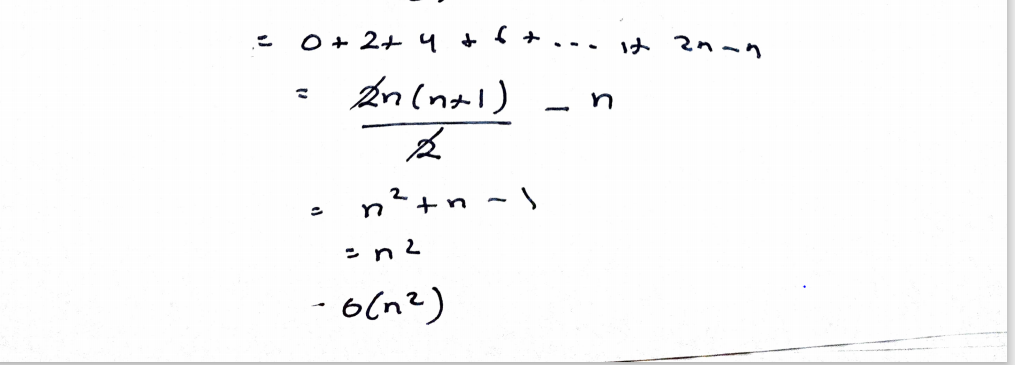
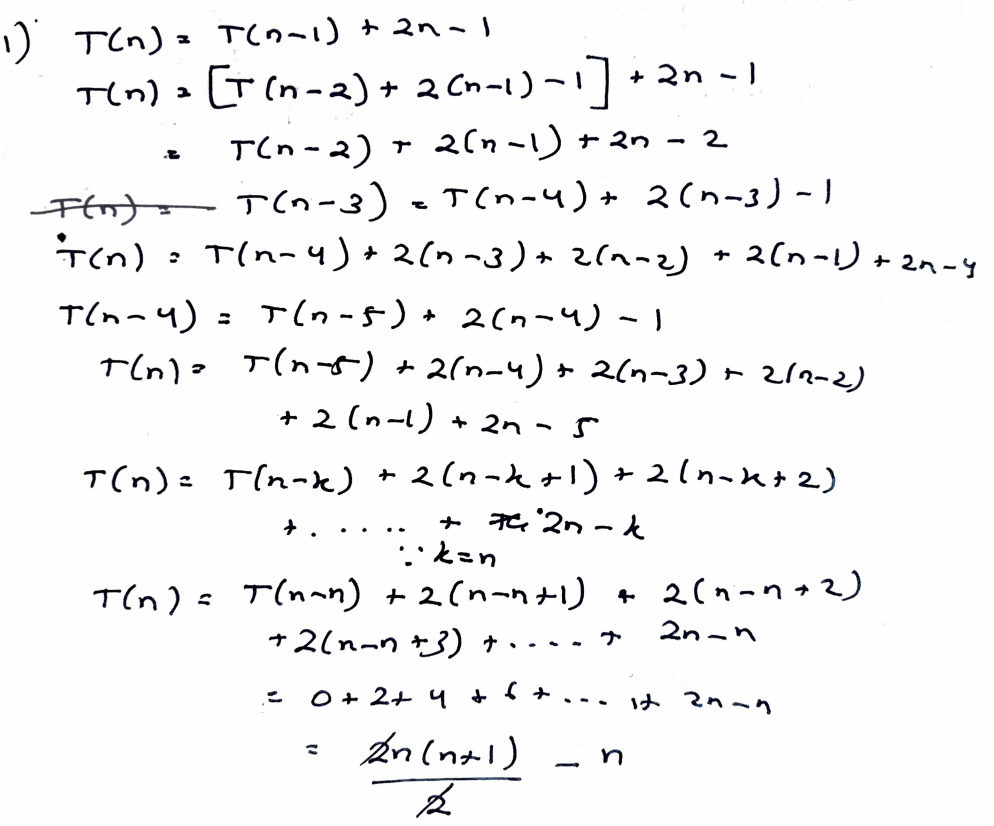


3rd Question a)

4.a) #include<stdio.h>

#include<math.h>

#include <cstdlib>

int maxRepeating(int arr[], int n, int k)

{

for (int i = 0; i< n; i++)

arr[arr[i]%k] += k;

int max = arr[0], result = 0;

for (int i = 1; i < n; i++)

{

if (arr[i] > max)

{

max = arr[i];

result = i;

}

}

for (int i = 0; i< n; i++)

arr[i] = arr[i]%k; \*/

printf( "\nMax repeating element:%d",result);

}

void printRandoms(int arr[],int lower, int upper,

int count)

{

int i;

for (i = 0; i < count; i++) {

int num = (rand() %(upper - lower + 1)) + lower;

printf("%d ", num);

arr[i]=num;

}

}

int main()

{

int n;

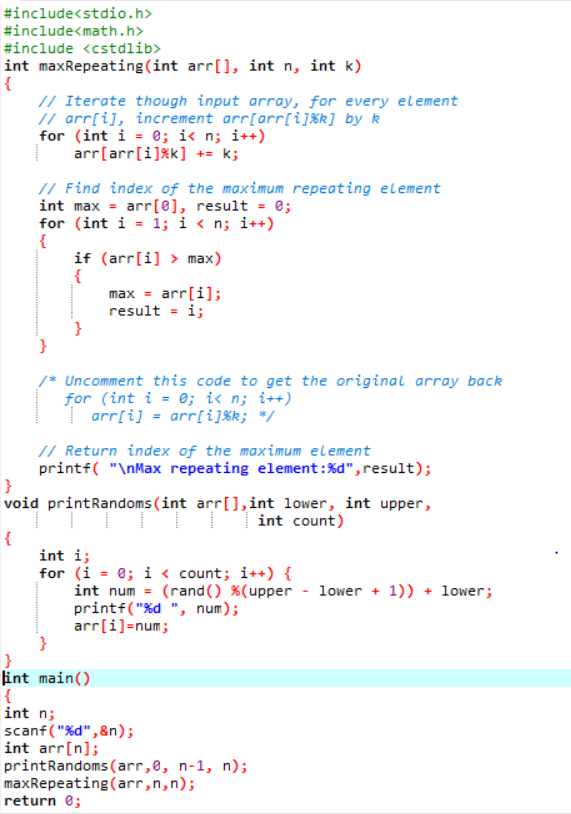
scanf("%d",&n);

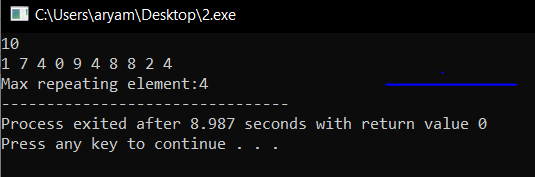
int arr[n];

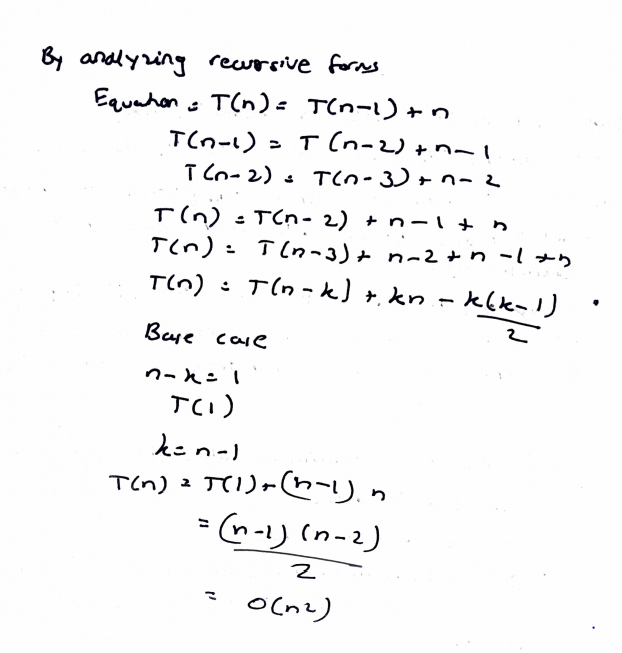
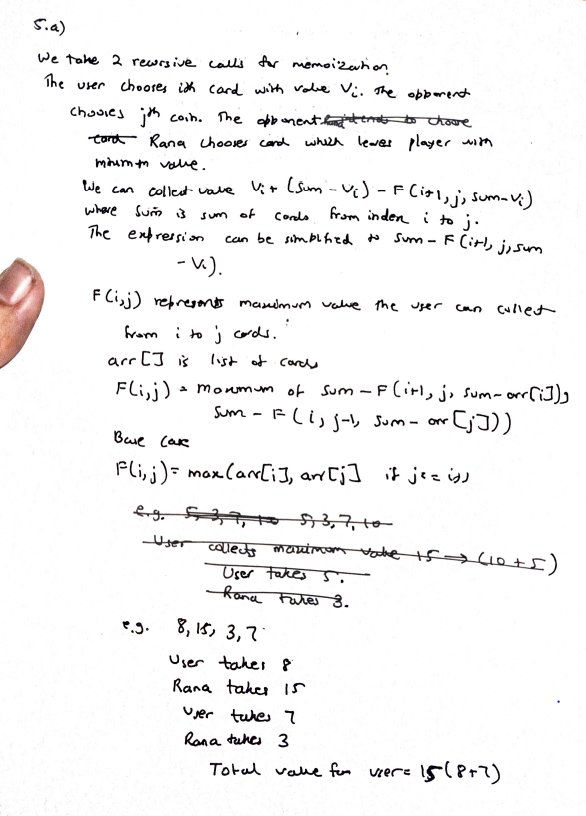
printRandoms(arr,0, n-1, n);

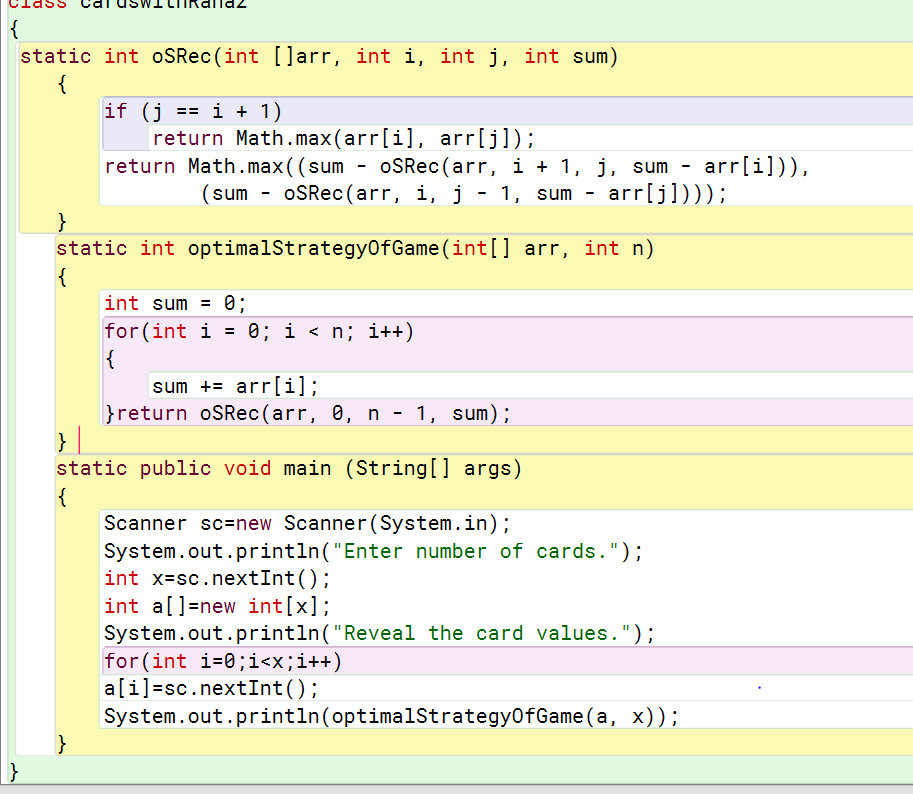
maxRepeating(arr,n,n);

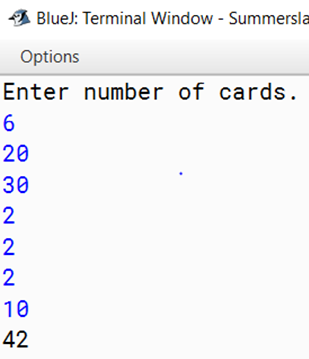
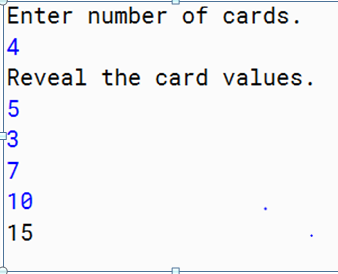
return 0;

}









5.a)

import java .io.\*;

import java.util.\*;

class cardswithRana

{

static int oSRec(int []arr, int i, int j, int sum)

{

if (j == i + 1)

return Math.max(arr[i], arr[j]);

return Math.max((sum - oSRec(arr, i + 1, j, sum - arr[i])),

(sum - oSRec(arr, i, j - 1, sum - arr[j]))); //Memoization

}

static int optimalStrategyOfGame(int[] arr, int n)

{

int sum = 0;

for(int i = 0; i < n; i++)

{

sum += arr[i];

}

return oSRec(arr, 0, n - 1, sum);

}

static public void main (String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of cards.");

int x=sc.nextInt();

int a[]=new int[x];

System.out.println("Reveal the card values.");

for(int i=0;i<x;i++)

a[i]=sc.nextInt();

System.out.println(optimalStrategyOfGame(a, x));

}

}

