Assignment on Algorithm Analysis and Arrays

Q1.) write a program to check a number is prime or not. calculate its time complexity with proper explaination.

code :-

```
import java.util.Scanner;
public class Q1{
  static boolean check(int n){
    int c=0;
    if(n==0||n==1)
      return false;
    else if(n==2)
      return true;
    else{
      for(int i=2; i<=Math.sqrt(n); i++){
        if(n%i==0)
          c++;
      }
    }
    if(c!=0)
      return false;
    else
      return true;
  }
  public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the number");
    int n=sc.nextInt();
```

Name:_____ Regd. Number:_____

System.out.println(check(n));

}

}

O/P :-

Enter the number

56

false.

Q.) write a program to find the binary equivalent of a decimal number. calculate its time complexity.

code :-

```
import java.util.Scanner;
public class Q2 {
static void binary(int n){
int [] b = new int [32];
int i = 0;
for(i=0; n>0; i++){
b[i] = n%2;
n = n/2;
}
for(int j= i-1; j>=0; j--)
system.out.print(b[j]);
}
public static void main(String[] ar
Scanner sc = new Scanner(System.
System.out.println("Enter the number to
convert from decimal to binary.n");
```

Name: _____ Regd. N

O/P 1. -

enter a number to reverse it

678

876

(4.) write a program to search an element using binary search and calculate its time complexity.

code 1 -

```
public class Q4{
 static boolean bsearch(int[] a, int v){
      int l=0;
      int h = a.length-1;
       int m;
  while (l<=h){
        m=(l+h)/2;
    if(a[m]==v){
       return true;
    }
  else if(a[m]<v){
       l=m+1; }
    else{
        h = m-1; }}
    return false;
}
public static void main(String[] args){
     int[] a ={5,7,8,9};
  System.out.println(bsearch(a,6));
}}
```

O/P 1 - false

```
int n = sc.nextInt();
binary(n);
}
}
```

O/P :-

Enter the number to convert from decimal to binary.

34

100010

(3.) write a program to find the reverse of a number and find its time complexity.

code :-

```
import java.util.Scanner;
public class Q3{
static int reverse(int n){
    int rc=0;
    while(n>0){
    rc=10*rc+(n%10);
    n=n/10;
    }return rc;
}
public static void main(String[] args){
Scanner sc=new Scanner(System.in);
System.out.println("enter a number to reverse it");
int n=sc.nextInt();
System.out.println(reverse(n));
}}
```

(5.) Given an array, write a program to rotate its element k number of times. Explain its time complexity.

code:-

```
public class Q5 {
  static void rotate(int a[], int k){
    int b[] = new int[a.length];
      int j = 0;
    for(int i=0; i<a.length; i++){
      if(i+k<=a.length-1){
        b[i] = a[i+k];
      }
      else{
        b[i] = a[j];
          j++;
      }}
    for(int i=0; i<b.length; i++){
      System.out.print(b[i]+" ");
    }}
  public static void main(String[] args){
    int a[] = {10,20,30,40,50,60,70,80,90};
    rotate(a,5);
  }}
```

O/P:-

60 70 80 90 10 20 30 40 50

6.) Given an array of positive and negative integers, write a program to find a contiguous sub array whose sum of element is maximum.

code :-

```
public class Q6 {
static int findSum(int[] a){
    int temp=0;
    int sum=0;
for(int i=0; i<a.length; i++){
    sum=sum+a[i];
    if(sum<0)
      sum=0;
    else if (sum>=temp)
      temp=sum;
  } return temp;
}
public static void main (String[] arg
int[] a={5,8,9,78,-45,90,-25,-89
          789,67};
System.out.println(findSum(a));
}}
```

O/P :-

856.

(7.) Given an array, write a program to arrange its elements in wave form such that odd elements are lesser than its neighbouring even elements.

code:-

```
public class Q7{
static void wave(int[] a){
    int temp=0;
for(int i=1; i<a.length; i+=2){
  if((i-1)>=0 && a[i]>a[i-1]){
      temp = a[i];
      a[i] = a[i-1];
      a[i-1] = temp;
  }
if((i+1)<a.length && a[i]>a[i+1]){
      temp = a[i];
      a[i] = a[i+1];
      a[i+1] = temp;
}
}
}
public static void main(String[] args){
  int[] a = {8,9,5,6,7,8,9,4,5};
  wave(a);
for(int i=0; i<a.length; i++)
  System.out.print(a[i]+" ");
}
```

Q1 :-

9 5 8 8 7 9 4 5

(8.) Given array of size N, containing elements from 0 to N-1. All values from 0 to N-1 are not in array or if they are not there then -1 is to take place. write a program to rearrange values of array so that value i stored at arr[i].

code :-

```
public class Q8{
    public static void main(String[] args){
    int[] a = {1,8,10,6};
    int max = 0;
    for(int i=0; i<a.length; i++){
        if(max<a[i])
            max = a[i];
    }
    int[] b = new int[max+1];
    for(int i=0; i<=max; i++){
        int count = 0;
        for(int j=0; j<a.length; j++){
            if(i==a[j]){
                b[i] = a[j];
                count++;
    }}
```

```
        if(count==0)
            s[i]=-1;
    }
for(int i=0;i<=max;i++){
    System.out.print(s[i]+" ");
}
}}
```

O/P :-

-1 -1 -1 -1 -1 -1 6 -1 8 -1 10

Q.) Write a recursive algorithm to solve Towers of Hanoi problem.

code :-

```
import java.util.Scanner;
Public class Q9{
Static void toh(int n, char a, char s,
                                char c){
  if(n>0){
   toh(n-1, a, c, s);
   System.out.println("move disc "+n+
                "from "+a+"to "+c);
   toh(n-1, s, a, c);
}}
Public static void main(String[] args){
   Scanner sc=new Scanner(System.in);
System.out.println("Enter the number
                of disc");
```

```
int n = sc.nextInt();
toh(n,'a','b','c');
}
}
```

o/p :-

Enter the number of dise
2
move disc 1 from a to b
move disc 2 from a to c
move disc 1 from b to c.

(Q.) write a recursive function to find GCD of two number. write the recurrence of the function and solve it for solution.

code :-

```
import java.util.Scanner;
public class GCD{
static int gcd(int a, int b){
if (a==0)
return b;
if(b==0)
return a;
if(a==b)
return a;
if(a>b)
return gcd(a-b,b);
return gcd(a,b-a);
}
```

Name:

```
public static void main (String[] args){
Scanner sc = new Scanner (System.in);
System.out.println ("Enter the two
                     numbers");
    int a = sc.nextInt();
    int b = sc.nextInt();
    System.out.println ("gcd is "+gcd(a,b));
  }
}
```

O/P :-

Enter the two numbers

56 78

gcd is 2

11) write a program to find all permutation of an integer list recursively.

code :-

```
class Allpermutation{
  private final int Arr[];
  private int Indexes[];
  private int Increase;
public Allpermutation (int arr[]){
    this.Arr = arr;
    this.Increase = -1;
   this.Indexes = new int [this.Arr.
                           length];
}
```

```
public void GetFirst(){
    this.Indexes = new int[this.length];
}
public for(int i=0; i<Indexes.length
                ++i){
    this.Indexes[i] = i;
}
this.Increase = 0;
this.output();
}
public boolean HasNext(){
return this.Increase != (this.Indexes
                length - 1);
}
public void GetNext(){
    if (this.increase == 0){
this.swap(this.Increase, this.Increa
    this.Increase += 1;
while(this.Increase < this.Indexes.leng
    && this.Indexes[this.Increase]
            > this.Indexes[this.Increa
    { ++this.Increase;
}}
```

PageNo:

(12.) write a recursive function to search an element using binary search. Analyse its time complexity.

code:-

```
public class Q12{
static boolean bsearch(int[] a, int l,
        int h, int v){

    if (l>h)
        return false;
    int m = (l+h)/2;
    if (a[m] == v)
        return true;
    else if (a[m] < v)
        return bsearch(a, m+1, h, v);
    else
        return bsearch(a, l, m-1, v);
}
public static void main(String[] args)
{
int[] a = {8, 9, 60, 90, 788};
System.out.println(bsearch(a,0,4,90));
}}
```

O/P:-

true

Q/S

16/5/22

Name: