

SORTING ANALYSIS CODE

```
import java.util.Arrays;
import java.util.Scanner;

class supplement{
    long getSelectionSortTime(int arr2[],int n)
    {
        long start=System.nanoTime();
        //Selection Sort
        for(int i=0;i<n;i++)
        {
            int min=arr2[i];
            int pos=i;
            for(int j=i;j<n;j++)
            {
                if(min>arr2[j])
                {
                    min=arr2[j];
                    pos=j;
                }
            }
            int temp=arr2[i];
            arr2[i]=arr2[pos];
            arr2[pos]=temp;
        }
        long end=System.nanoTime();
        return (end-start);
    }
    long getBubbleSortTime(int arr1[],int n)
    {
        //Bubble sort
        long start=System.nanoTime();
        for(int i=0;i<n-1;i++)
        {
            for(int j=0;j<n-1-i;j++)
            {
                if(arr1[j]>arr1[j+1])
                {
                    int temp=arr1[j];
                    arr1[j]=arr1[j+1];
                    arr1[j+1]=temp;
                }
            }
        }
        long end=System.nanoTime();
        return end-start;
    }
}
```

```
int[] reverseArray(int arr[],int n)
{
    for(int i=0;i<=n/2;i++)
    {
        int temp=arr[i];
        arr[i]=arr[n-i-1];
        arr[n-1-i]=temp;
    }
    return arr;
}

public class assignment {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];    //Initialising an array of size 100
        int arr1[]=new int[n];
        for(int i=0;i<100;i++)
        {
            int k=(int)(Math.random()*10000);    //Assigning random values
            if(k<1000)
            {
                i--;
                continue;
            }
            arr[i]=k;
            arr1[i]=arr[i];
        }
        supplement x1=new supplement();    //Creating an object of supplement
class
        double timeElapsed=0,avg1=0,avg2=0,avg3=0;
        for(int k=1;k<=10;k++)
        {
            timeElapsed = x1.getBubbleSortTime(arr, n);
            timeElapsed/=1000;
            avg1=avg1+timeElapsed;
            //System.out.println("Bubble sort time elapsed = "+timeElapsed);
            Arrays.sort(arr1);
            timeElapsed = x1.getBubbleSortTime(arr1, n);
            timeElapsed/=1000;
            avg2=avg2+timeElapsed;
            //System.out.println("Bubble sort in ascending order = "+
timeElapsed);
            arr1=x1.reverseArray(arr1, n);
            timeElapsed = x1.getBubbleSortTime(arr1, n);
            timeElapsed/=1000;
            avg3=avg3+timeElapsed;
```

```
        //System.out.println("Bubble sort in descending order = "+
timeElapsed);
    }

    avg1=avg1/10;
    avg2=avg2/10;
    avg3=avg3/10;
    System.out.println("Bubble Sort time = "+avg1);
    System.out.println("Bubble Sort time Ascending order = "+avg2);
    System.out.println("Bubble Sort time Descending order= "+avg3);

    //Selection Sort
    avg1=0;avg2=0;avg3=0;
    for(int k=1;k<=10;k++)
    {
        timeElapsed = x1.getSelectionSortTime(arr, n);
        timeElapsed/=1000;
        avg1+=timeElapsed;
        //System.out.println("Selection sort time elapsed =
"+timeElapsed);
        Arrays.sort(arr1);
        timeElapsed = x1.getSelectionSortTime(arr1, n);
        timeElapsed/=1000;
        avg2+=timeElapsed;
        //System.out.println("Selection sort in ascending order = "+
timeElapsed);
        arr1=x1.reverseArray(arr1, n);
        timeElapsed = x1.getSelectionSortTime(arr1, n);
        timeElapsed/=1000;
        avg3+=timeElapsed;
        //System.out.println("Selection sort in descending order = "+
timeElapsed);
    }
    avg1=avg1/10;
    avg2=avg2/10;
    avg3=avg3/10;
    System.out.println("Selection Sort time = "+avg1);
    System.out.println("Selection Sort time Ascending order = "+avg2);
    System.out.println("Selection Sort time Descending order= "+avg3);

}
}
```

OUTPUT

```
200
Bubble Sort time = 185.15
Bubble Sort time Ascending order = 109.81999999999998
Bubble Sort time Descending order= 145.20000000000002
Selection Sort time = 150.2
Selection Sort time Ascending order = 146.07
Selection Sort time Descending order= 128.48
```

TABLE

S No.	Sorting Algorithm	Number of elements in the array	Average Time (in ms) Random order	Average Time (in ms) Ascending Order	Average Time (in ms) Descending Order
1.	Bubble Sort	200	185.15	109.82	145.20
2.	Selection Sort	200	150.20	146.07	128.48

PATTERN CODE

```
import java.util.Scanner;

public class pattern {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int spaces=0;
        for(int i=n;i>0;i--)
        {
            for(int j=0;j<spaces;j++)
            {
                System.out.print(" ");
            }
            for(int j=i;j>0;j--)
            {
                System.out.print("* ");
            }
            System.out.println();
            spaces++;
        }

        spaces-=2;
        for(int i=1;i<n;i++)
        {
            for(int j=0;j<spaces;j++)
            {
                System.out.print(" ");
            }
            for(int j=0;j<=i;j++)
            {
                System.out.print("* ");
            }
            System.out.println();
            spaces--;
        }
    }
}
```

OUTPUT

```
5
* * * * *
 * * * *
  * * *
   * *
    *
   * *
  * * *
 * * * *
* * * * *
```

ARRAY PROBLEM CODE

```
import java.util.Scanner;
class sup{
    void check(int arr[],int n) //Function to Check True or False
    {
        if(arr[0]!=0)
        {
            System.out.println("False");
            return;
        }
        for(int i=1;i<n;i++)
        {
            if(arr[i]%i!=0)
            {
                System.out.println("False");
                return;
            }
        }
        System.out.println("True");
    }
}

public class arrayproblem {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
        sup x1=new sup(); //Creating object of class sup
        x1.check(arr,n); //Accessing check()
    }
}
```

OUTPUT

```
6
0 2 6 21 16 15
True
```

```
6
0 2 3 21 16 15
False
```