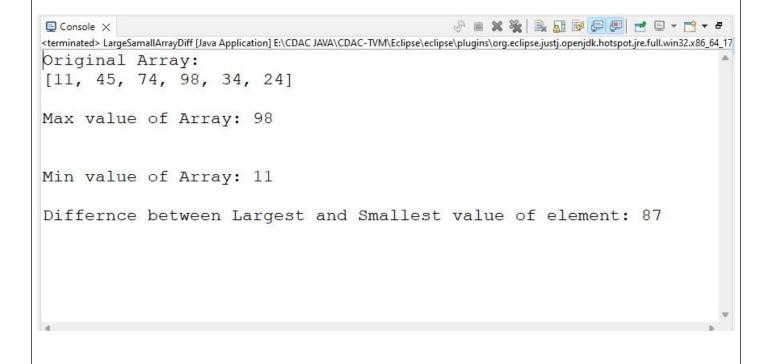
Name: Manish Kumavat PRN NO.: 220960920041

## **Exam**

1. Write a program to find the difference between the largest and smallest values in an array of integers.

```
package com.arraylist.main;
import java.util.Arrays;
public class LargeSamallArrayDiff {
   public static void main(String[] args)
     int arr[] = {11,45,74,98,34,24};
     System.out.println("Original Array:");
     System.out.println(Arrays.toString(arr));
     System.out.println("");
     int max=arr[0];
     for(int i=1;i<arr.length;i++)</pre>
         if(arr[i]>max)
            max=arr[i];
     System.out.println("Max value of Array: "+max);
     System.out.println("");
     int min=arr[0];
     for(int i=1;i<arr.length;i++)</pre>
         if (arr[i] < min)</pre>
            min=arr[i];
```

```
}
}
System.out.println("");
System.out.println("Min value of Array: "+min);
System.out.println("");
System.out.println("Differnce between Largest and
Smallest value of element: "+(max-min));
}
}
```



2. Write a C program to create a parent process which terminates after the child finishes printing the contents of array.

```
-(kali⊕kali)-[~/exam]
_$ cat proc_child.c
#include<stdio.h>
#include<svs/wait.h>
#include<stdlib.h>
#include<unistd.h>
void main()
 pid_t id;
 id=fork();
if(id>0)
printf("Parent Started Executing \n");
printf("Waiting for child to finish \n");
wait(NULL);
printf("Parent Exiting \n");
else
printf("Child Executing \n");
 int arr[]={1,2,3,4,5};
 int i;
 for(i=0;i<5;i++)
 sleep(5);
printf("%d \n",arr[i]);
printf("child Finished \n");
 exit(0);
```

```
(kali® kali)-[~/exam]
$ vim proc_child.c

(kali® kali)-[~/exam]
$ gcc proc_child.c -o proc_child.out

(kali® kali)-[~/exam]
$ ./proc_child.out

Parent Started Executing
Child Executing
Waiting for child to finish
1
2
3
4
5
child Finished
Parent Exiting
```