PRN: 49

Name: Vishal Murudkar

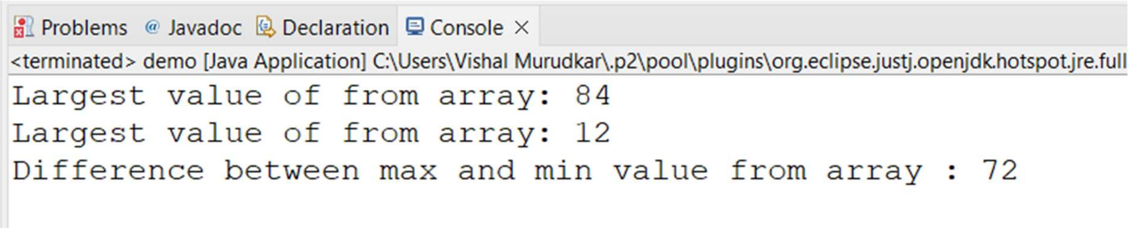
1. Write a program to find the difference between the largest and smallest values in an array of integers.

Answer:

Code:

```
package com.demo.main;

import java.util.Arrays;

public class demo {

        public static void main(String[] args) {

                int[] arr = {15,12,84,65,75,14};

                int max = arr[0];
                int min = arr[0];
                for(int i = 1; i < arr.length; i++)
                {
                        if(arr[i] > max)
                                max = arr[i];
                        else if(arr[i] < min)
                                min = arr[i];
                }
                int diff=max-min;
                System.out.println("Largest value of from array: "+max);
                System.out.println("Largest value of from array: "+min);
                System.out.println("Difference between max and min value from array : "+diff);
        }
}
```

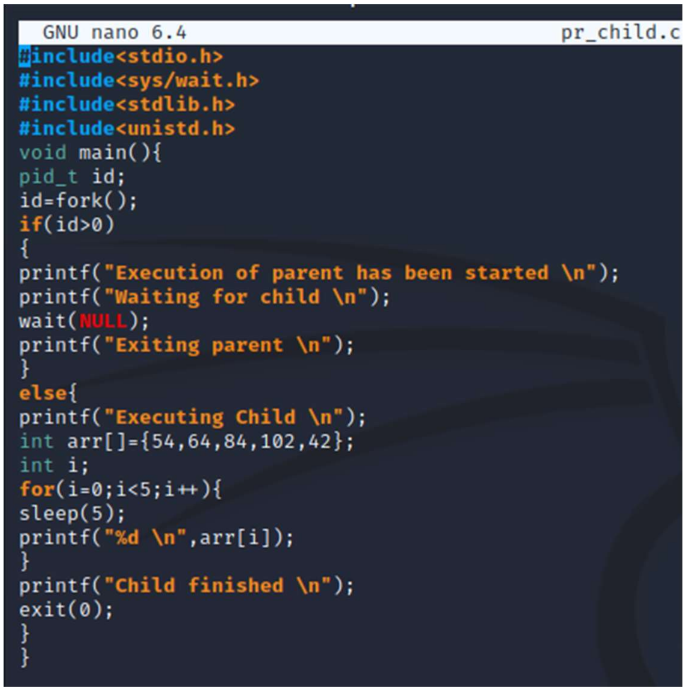
Output:



```
<terminated> demo [Java Application] C:\Users\Vishal Murudkar\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Largest value of from array: 84
Largest value of from array: 12
Difference between max and min value from array : 72
```

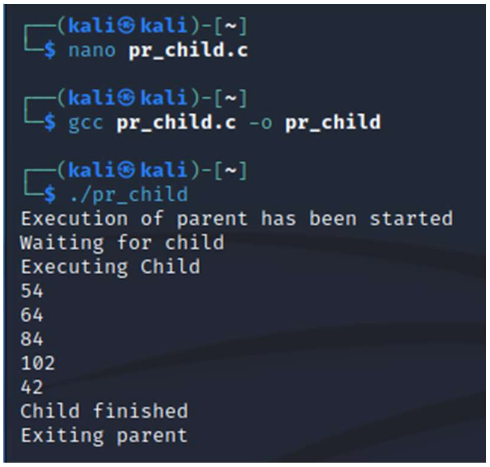
2. Write a C program to create a parent process which terminates after the child finishes printing the contents of array.

Answer:



```
  GNU nano 6.4                                   pr_child.c
#include<stdio.h>
#include<sys/wait.h>
#include<stdlib.h>
#include<unistd.h>
void main(){
pid_t id;
id=fork();
if(id>0)
{
printf("Execution of parent has been started \n");
printf("Waiting for child \n");
wait(NULL);
printf("Exiting parent \n");
}
else{
printf("Executing Child \n");
int arr[]={54,64,84,102,42};
int i;
for(i=0;i<5;i++){
sleep(5);
printf("%d \n",arr[i]);
}
printf("Child finished \n");
exit(0);
}
}
```

Output:



```
┌──(kali㉿kali)-[~]
└─$ nano pr_child.c

┌──(kali㉿kali)-[~]
└─$ gcc pr_child.c -o pr_child

┌──(kali㉿kali)-[~]
└─$ ./pr_child
Execution of parent has been started
Waiting for child
Executing Child
54
64
84
102
42
Child finished
Exiting parent
```