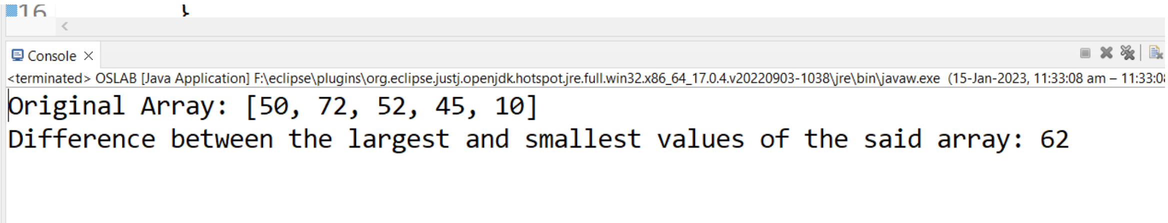
OS LAB EXAM

Tejas Khadke 220960920038

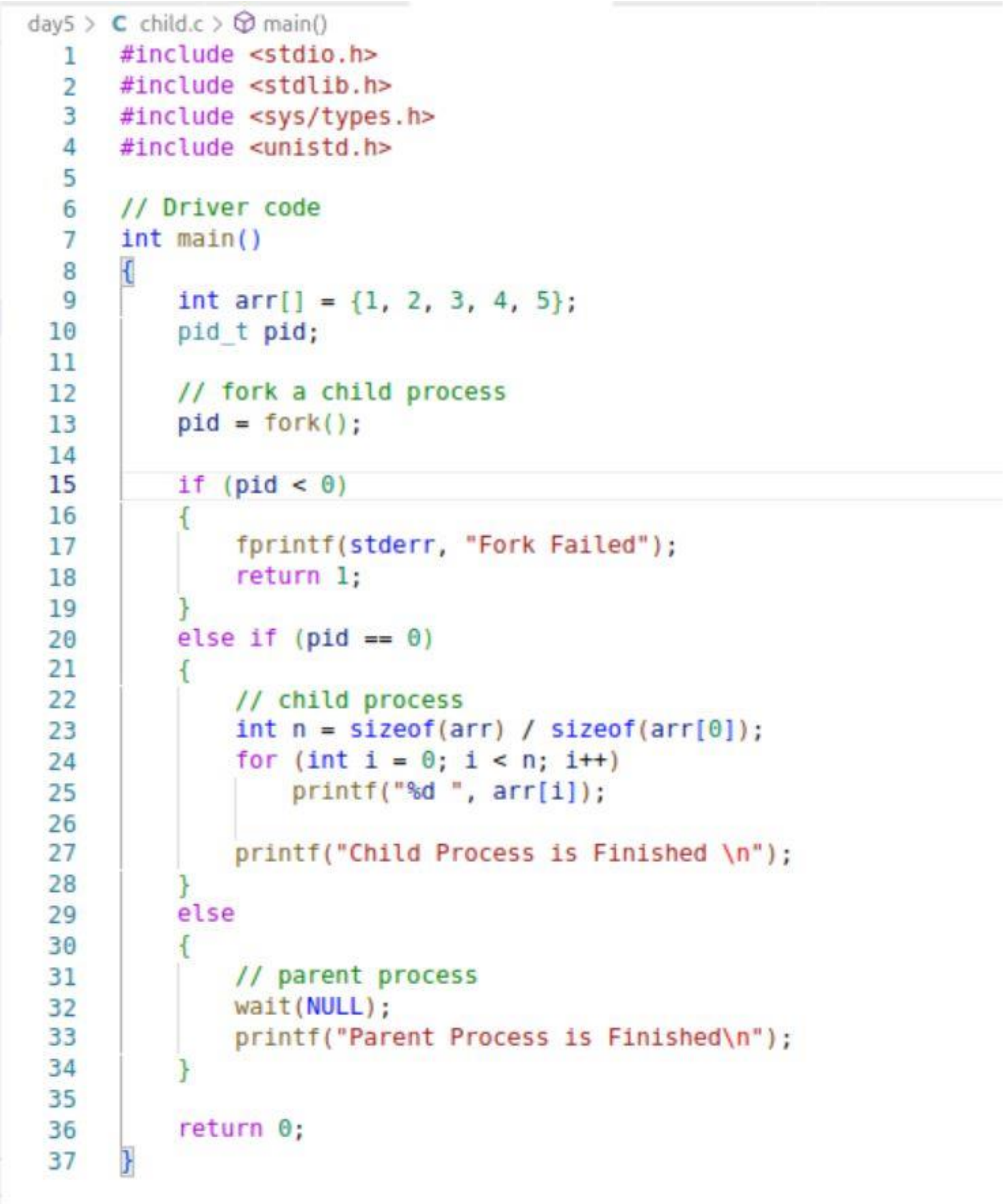
1)

```
package com.os.main;
import java.util.Arrays;
public class OSLAB {
    public static void main(String[] args)
     {
        int[] array_nums = {50, 72, 52, 45, 
10};
       System.out.println("Original Array: 
"+Arrays.toString(array_nums));
       int max_val = array_nums[0];
       int min = array_nums[0];
       for(int i = 1; i < array_nums.length; 
i++)
       {
           if(array_nums[i] > max_val)
               max_val = array_nums[i];
           else if(array_nums[i] < min)
               min = array_nums[i];
       }
       System.out.println("Difference between 
the largest and smallest "
               + "values of the said array: 
"+(max_val-min));
     }
}
```

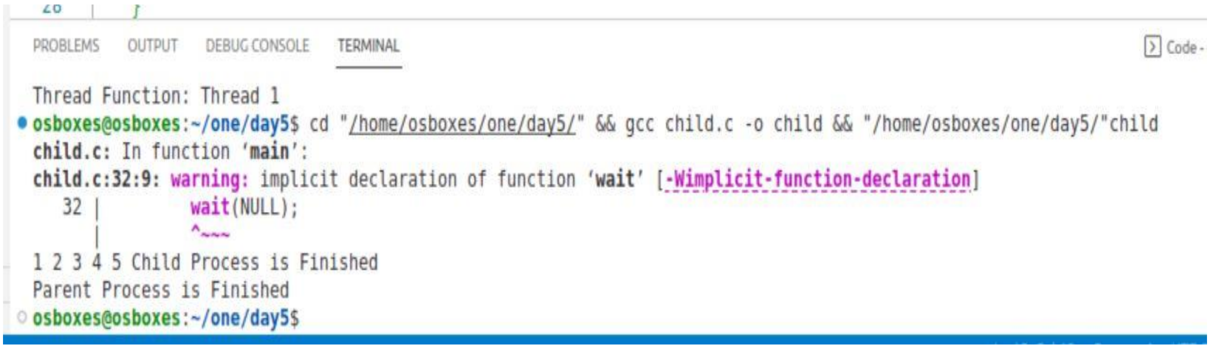


```
<terminated> OSLAB [Java Application] F:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe  (15-Jan-2023, 11:33:08 am – 11:33:0
Original Array: [50, 72, 52, 45, 10]
Difference between the largest and smallest values of the said array: 62
```

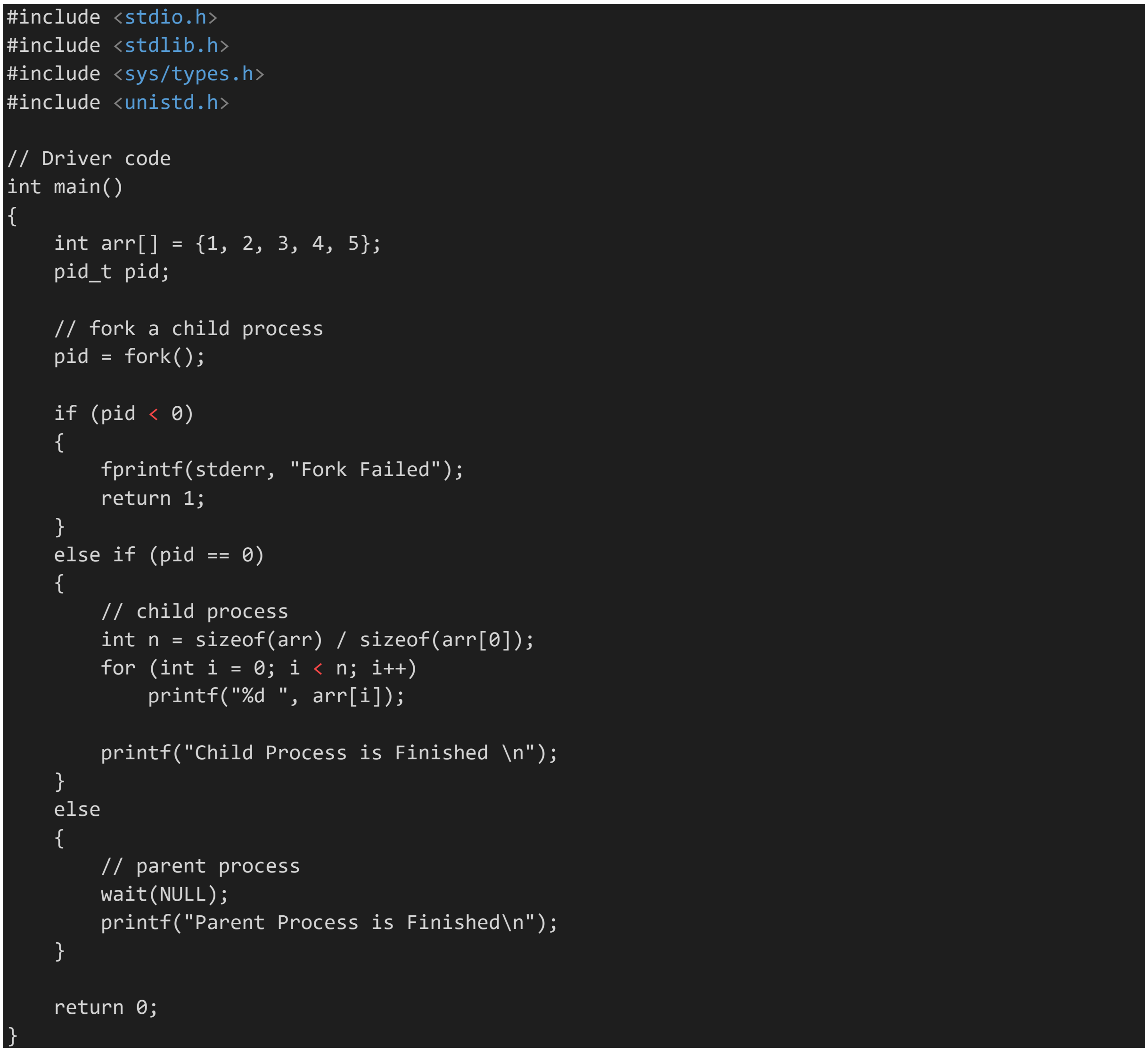
2)



```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

// Driver code
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    pid_t pid;

    // fork a child process
    pid = fork();

    if (pid < 0)
    {
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0)
    {
        // child process
        int n = sizeof(arr) / sizeof(arr[0]);
        for (int i = 0; i < n; i++)
            printf("%d ", arr[i]);

        printf("Child Process is Finished \n");
    }
    else
    {
        // parent process
        wait(NULL);
        printf("Parent Process is Finished\n");
    }

    return 0;
}
```



```
Thread Function: Thread 1
osboxes@osboxes:~/one/day5$ cd "/home/osboxes/one/day5/" && gcc child.c -o child && "/home/osboxes/one/day5/"child
child.c: In function ‘main’:
child.c:32:9: warning: implicit declaration of function ‘wait’ [-Wimplicit-function-declaration]
   32 |         wait(NULL);
      |         ^~~~
1 2 3 4 5 Child Process is Finished
Parent Process is Finished
osboxes@osboxes:~/one/day5$
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

// Driver code
int main()
{
    int arr[] = {1, 2, 3, 4, 5};
    pid_t pid;

    // fork a child process
    pid = fork();

    if (pid < 0)
    {
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0)
    {
        // child process
        int n = sizeof(arr) / sizeof(arr[0]);
        for (int i = 0; i < n; i++)
            printf("%d ", arr[i]);

        printf("Child Process is Finished \n");
    }
    else
    {
        // parent process
        wait(NULL);
        printf("Parent Process is Finished\n");
    }

    return 0;
}
```