Name: Nikhil Gupta
Roll No.: 20051523

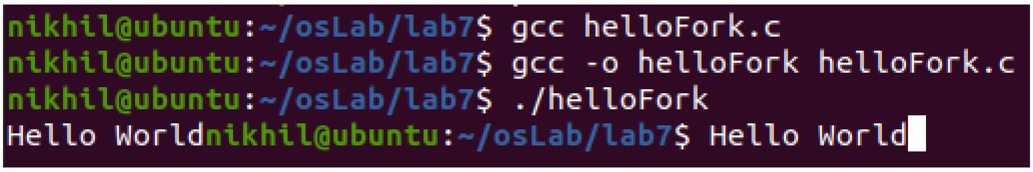
INDEX

LAB ASSIGNMENT 7

Ques1. Predict the Output of the following program

```c
// Predict Output of Following Program

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    fork();
    printf("Hello World");
    return 0;
}
```

Output



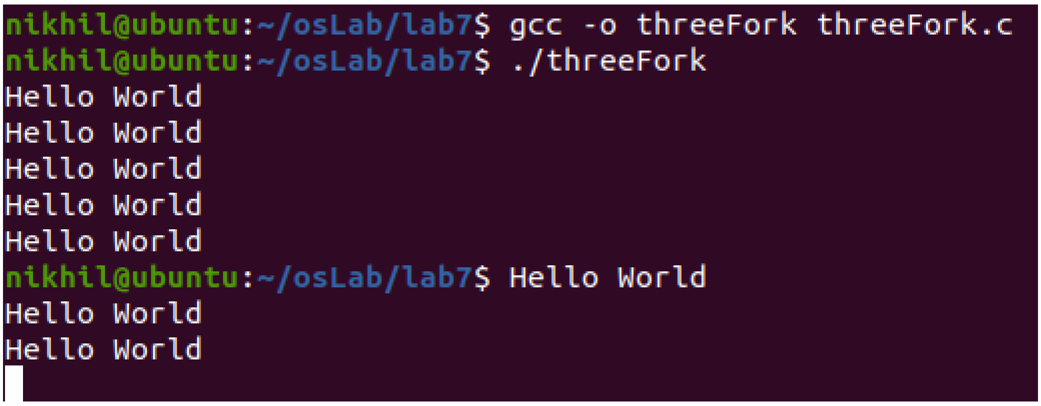
```
nikhil@ubuntu:~/osLab/lab7$ gcc helloFork.c
nikhil@ubuntu:~/osLab/lab7$ gcc -o helloFork helloFork.c
nikhil@ubuntu:~/osLab/lab7$ ./helloFork
Hello Worldnikhil@ubuntu:~/osLab/lab7$ Hello World
```

Ques 2. Calculate number of times hello is printed

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

int main()
{
    fork();
    fork();
    fork();
    printf("Hello World\n");
    return 0;
}
```

Output

Hello will be printed 8 times

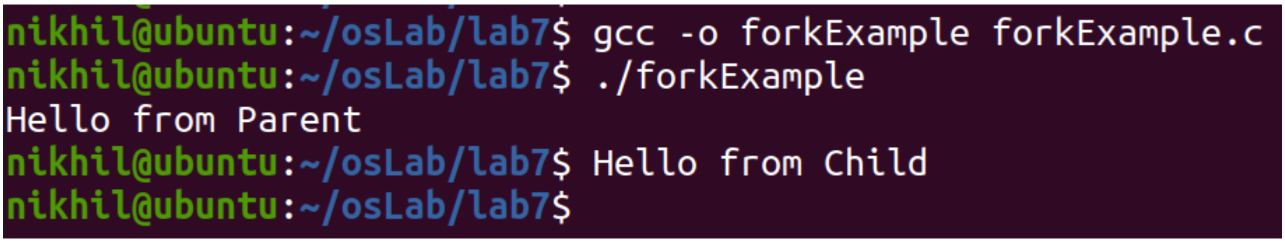


```
nikhil@ubuntu:~/osLab/lab7$ gcc -o threeFork threeFork.c
nikhil@ubuntu:~/osLab/lab7$ ./threeFork
Hello World
Hello World
Hello World
Hello World
Hello World
nikhil@ubuntu:~/osLab/lab7$ Hello World
Hello World
Hello World
```

Ques 3. Predict the Output of the following program

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

void forkexample()
{
    if (fork() == 0)
    {
        printf("Hello from Child\n");
    }
    else
    {
        printf("Hello from Parent\n");
    }
}
int main()
{
    forkexample();
    return 0;
}
```

Output

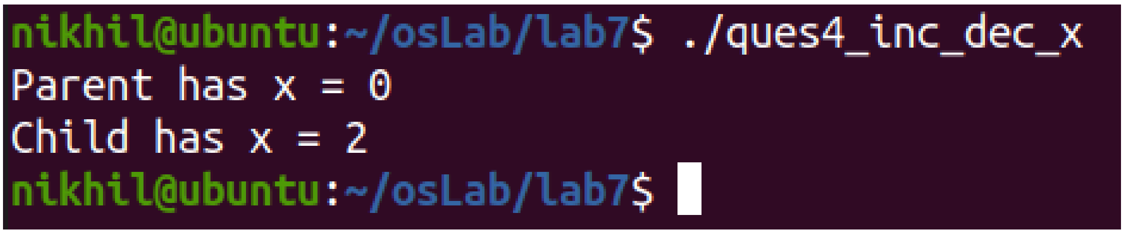


```
nikhil@ubuntu:~/osLab/lab7$ gcc -o forkExample forkExample.c
nikhil@ubuntu:~/osLab/lab7$ ./forkExample
Hello from Parent
nikhil@ubuntu:~/osLab/lab7$ Hello from Child
nikhil@ubuntu:~/osLab/lab7$
```

Ques 4. Predict the Output of the following

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void forkexample()
{
    int x = 1;
    if (fork() == 0)
        printf("Child has x = %d\n", ++x);
    else
        printf("Parent has x = %d\n", --x);
}
int main()
{
    forkexample();
    return 0;
}
```

Output

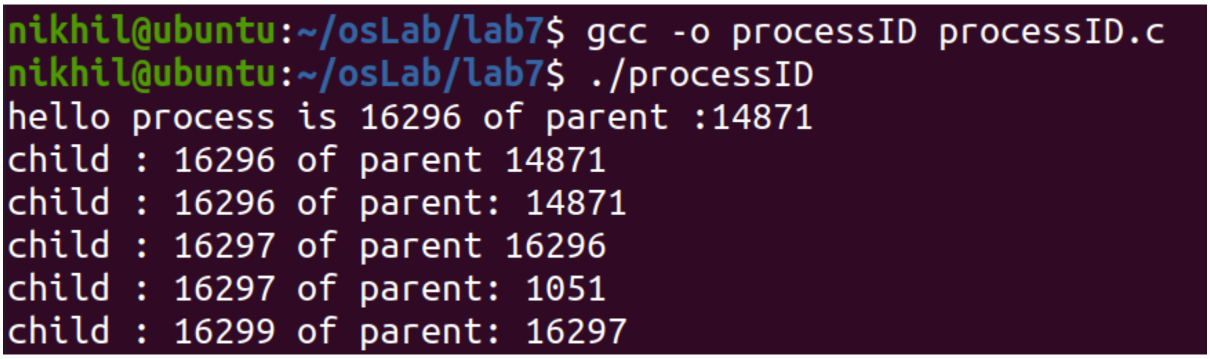


```
nikhil@ubuntu:~/osLab/lab7$ ./ques4_inc_dec_x
Parent has x = 0
Child has x = 2
nikhil@ubuntu:~/osLab/lab7$
```

Ques 5 Write a Program to find process id and parent process id.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    printf("hello process is %d of parent :%d\n",
getpid(), getppid());
    fork();
    printf("child : %d of parent %d \n", getpid(),
getppid());
    fork();
    printf("child : %d of parent: %d\n", getpid(),
getppid());
    return 0;
}
```

Output

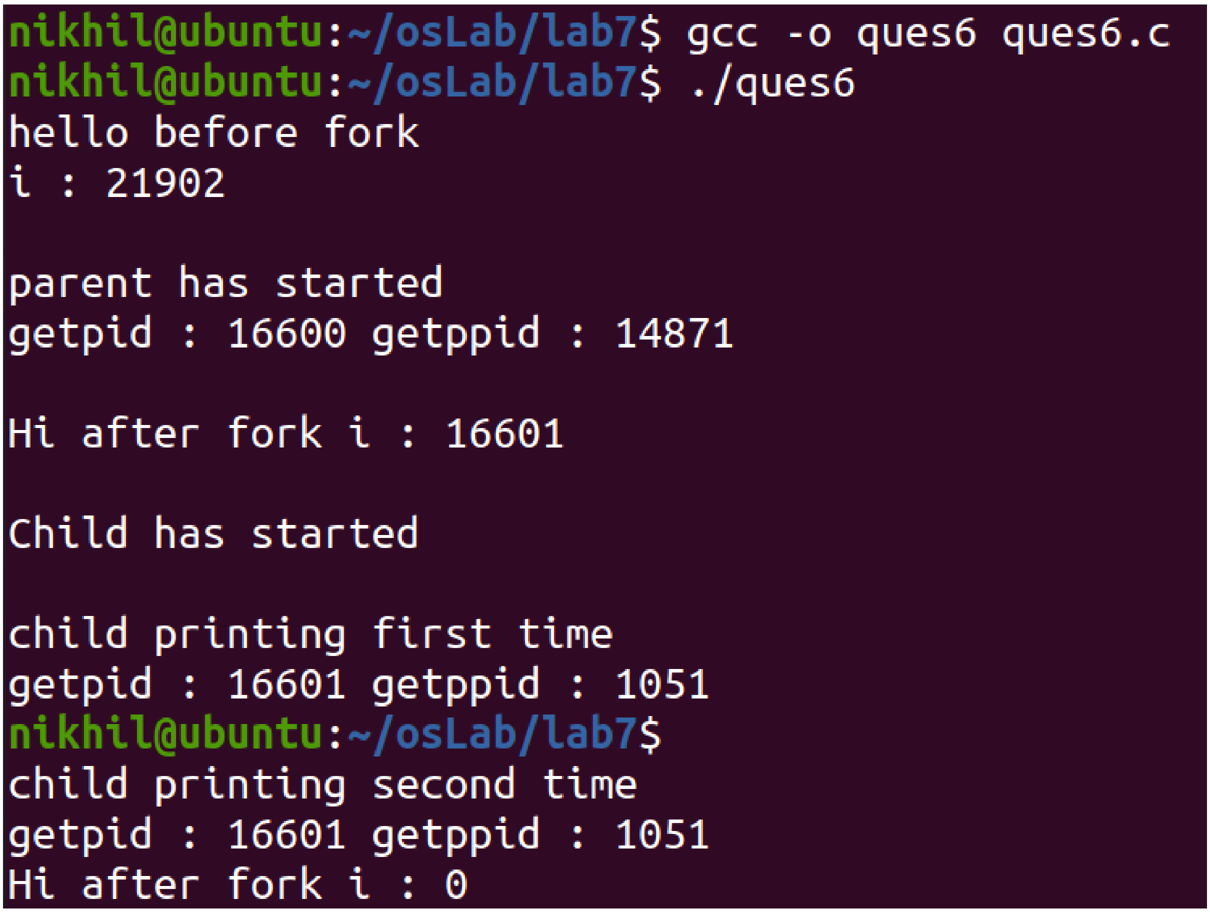


```
nikhil@ubuntu:~/osLab/lab7$ gcc -o processID processID.c
nikhil@ubuntu:~/osLab/lab7$ ./processID
hello process is 16296 of parent :14871
child : 16296 of parent 14871
child : 16296 of parent: 14871
child : 16297 of parent 16296
child : 16297 of parent: 1051
child : 16299 of parent: 16297
```

Ques 6. Write a program to fork a child and print the process id of parent and child process

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    int i;
    printf("hello before fork \n");
    printf("i : %d\n", i);
    i = fork();
    printf("\n");
    if (i == 0)
    {
        printf("Child has started\n\n");
        printf("child printing first time \n");
        printf("getpid : %d getppid : %d \n", getpid(), getppid());
        sleep(5);
        printf("\nchild printing second time \n");
        printf("getpid : %d getppid : %d \n", getpid(), getppid());
    }
    else
    {
        printf("parent has started\n");
        printf("getpid : %d getppid : %d \n", getpid(), getppid());
        printf("\n");
    }
    printf("Hi after fork i : %d\n", i);
    return 0;
}
```

Output

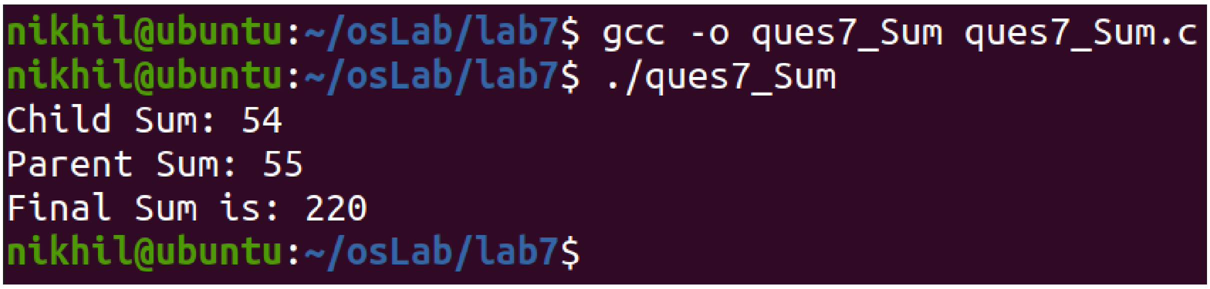


```
nikhil@ubuntu:~/osLab/lab7$ gcc -o ques6 ques6.c
nikhil@ubuntu:~/osLab/lab7$ ./ques6
hello before fork
i : 21902

parent has started
getpid : 16600 getppid : 14871

Hi after fork i : 16601

Child has started

child printing first time
getpid : 16601 getppid : 1051
nikhil@ubuntu:~/osLab/lab7$
child printing second time
getpid : 16601 getppid : 1051
Hi after fork i : 0
```

Ques 7. Program to do Sum of numbers from 1 to 10, by dividing the job into two processes (parent and one child) in C Programming.

```c
#include <stdio.h>
#include <unistd.h>
int main()
{
    int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int sum1 = 0, sum2 = 0, sum = 0;
    int id;
    id = fork();
    if (id == 0)
    {
        for (int i = 0; i < 10; i++)
        {
            sum1 += a[i];
            sum += sum1;
        }
        printf("Parent Sum: %d\n", sum1);
    }
    else
    {
        for (int i = 1; i < 10; i++)
        {
            sum2 += a[i];

            sum += sum2;
        }
        printf("Child Sum: %d\n", sum2);
    }
    if (id == 0)
    {
        printf("Final Sum is: %d\n", sum);
    }
    return 0;
}
```

Output



```
nikhil@ubuntu:~/osLab/lab7$ gcc -o ques7_Sum ques7_Sum.c
nikhil@ubuntu:~/osLab/lab7$ ./ques7_Sum
Child Sum: 54
Parent Sum: 55
Final Sum is: 220
nikhil@ubuntu:~/osLab/lab7$
```