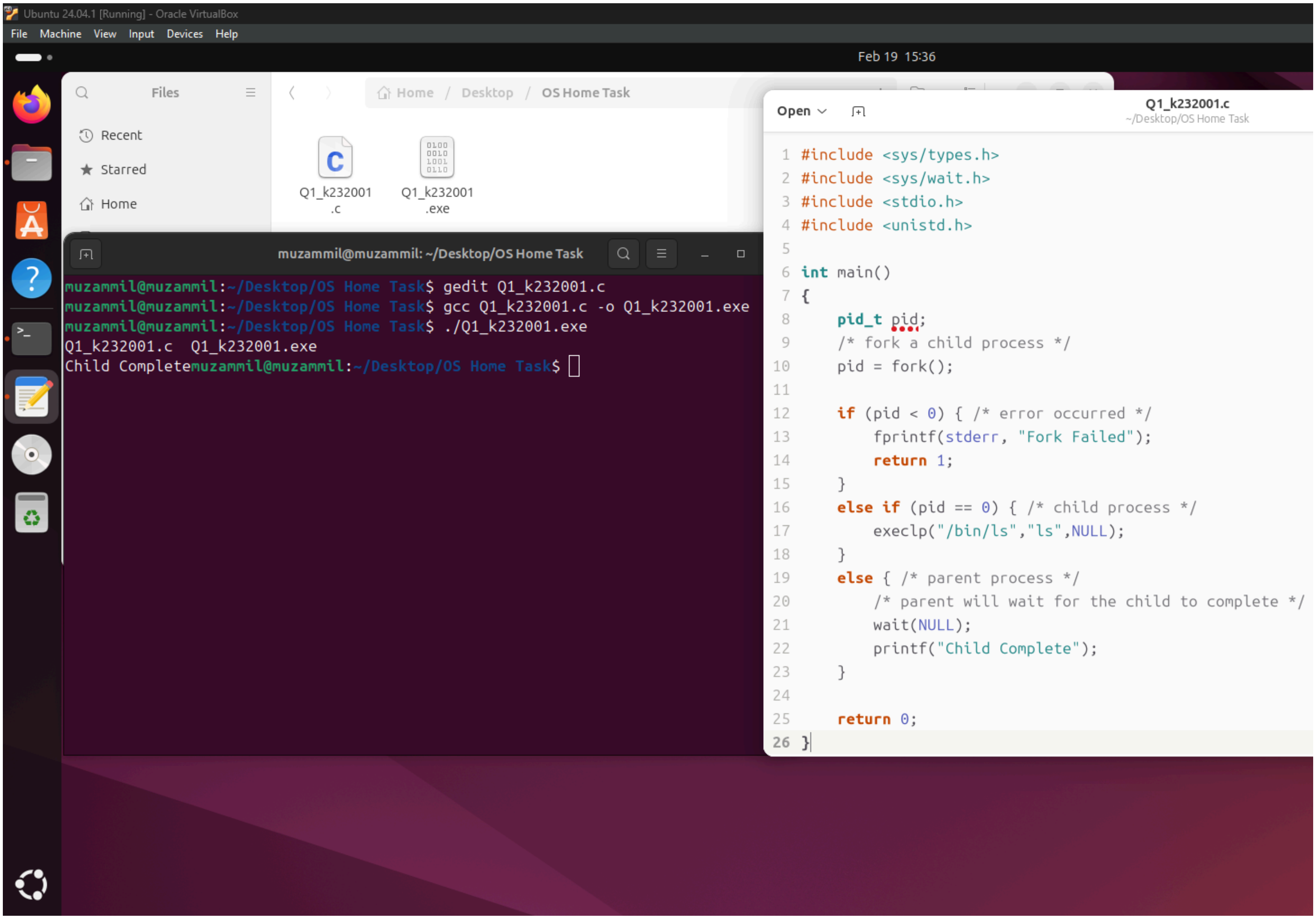
Operating Systems

HOMETASK#01

23K-2001

BCS-4J

Q1:

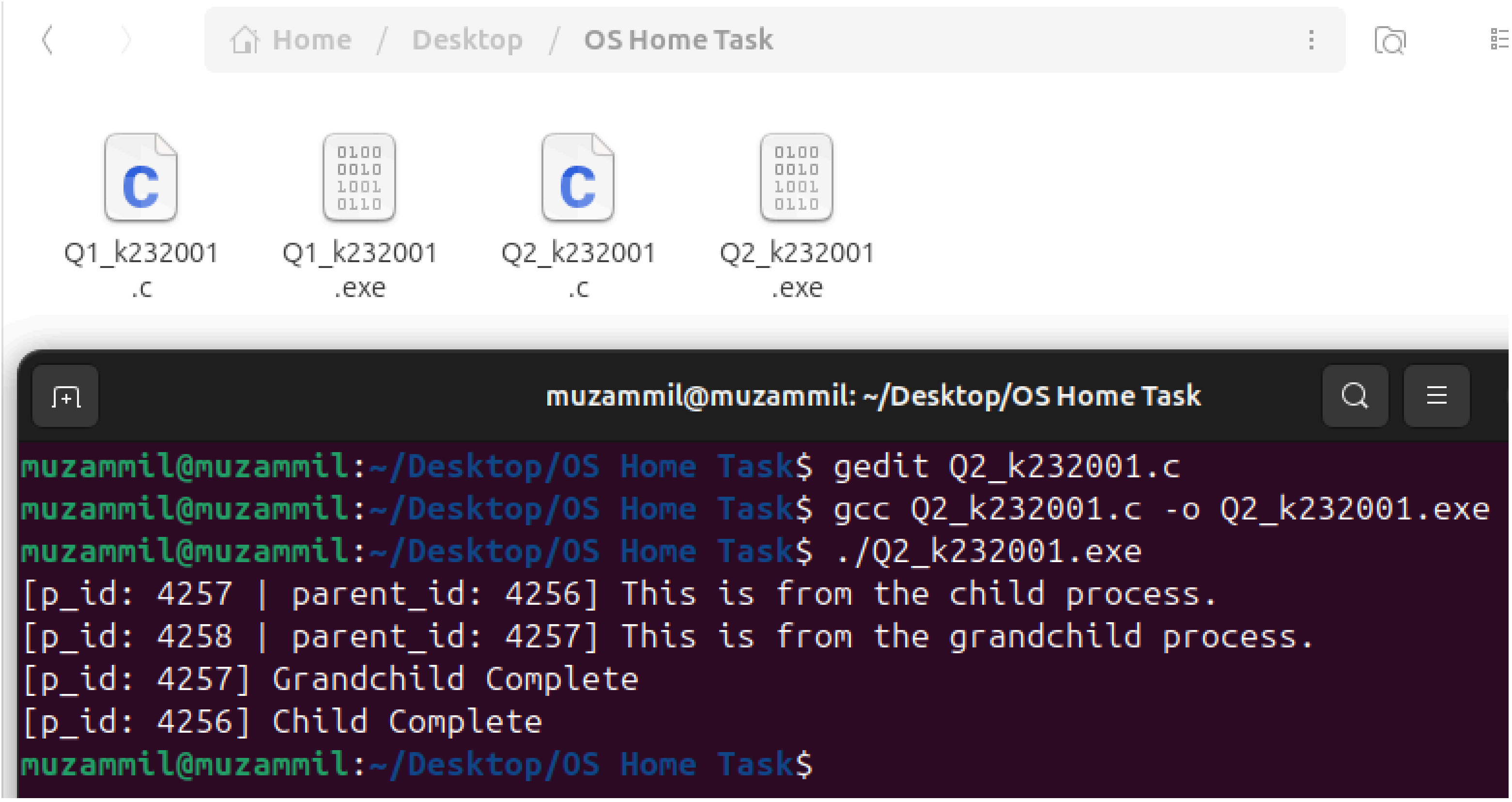


Ubuntu 24.04.1 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Feb 19 15:36
Files
Recent
Starred
Home
Home / Desktop / OS Home Task
Q1_k232001.c
Q1_k232001.exe
muzammil@muzammil: ~/Desktop/OS Home Task
Open
Q1_k232001.c
~/Desktop/OS Home Task

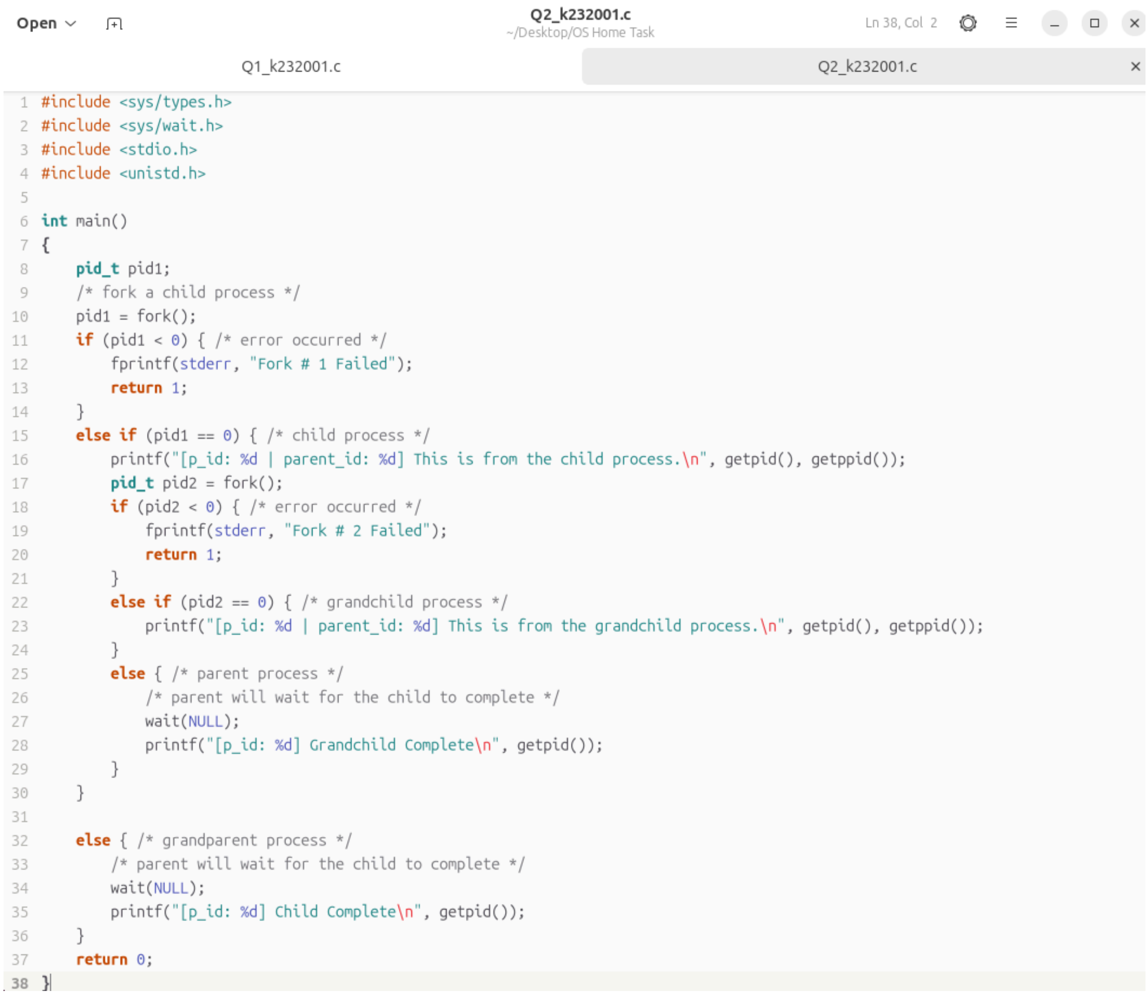
```
muzammil@muzammil:~/Desktop/OS Home Task$ gedit Q1_k232001.c
muzammil@muzammil:~/Desktop/OS Home Task$ gcc Q1_k232001.c -o Q1_k232001.exe
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q1_k232001.exe
Q1_k232001.c  Q1_k232001.exe
Child Completemuzammil@muzammil:~/Desktop/OS Home Task$ 
```

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

int main()
{
    pid_t pid;
    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls","ls",NULL);
    }
    else { /* parent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
    }

    return 0;
}
```

Q2:


Home / Desktop / OS Home Task
Q1_k232001.c
Q1_k232001.exe
Q2_k232001.c
Q2_k232001.exe
muzammil@muzammil: ~/Desktop/OS Home Task

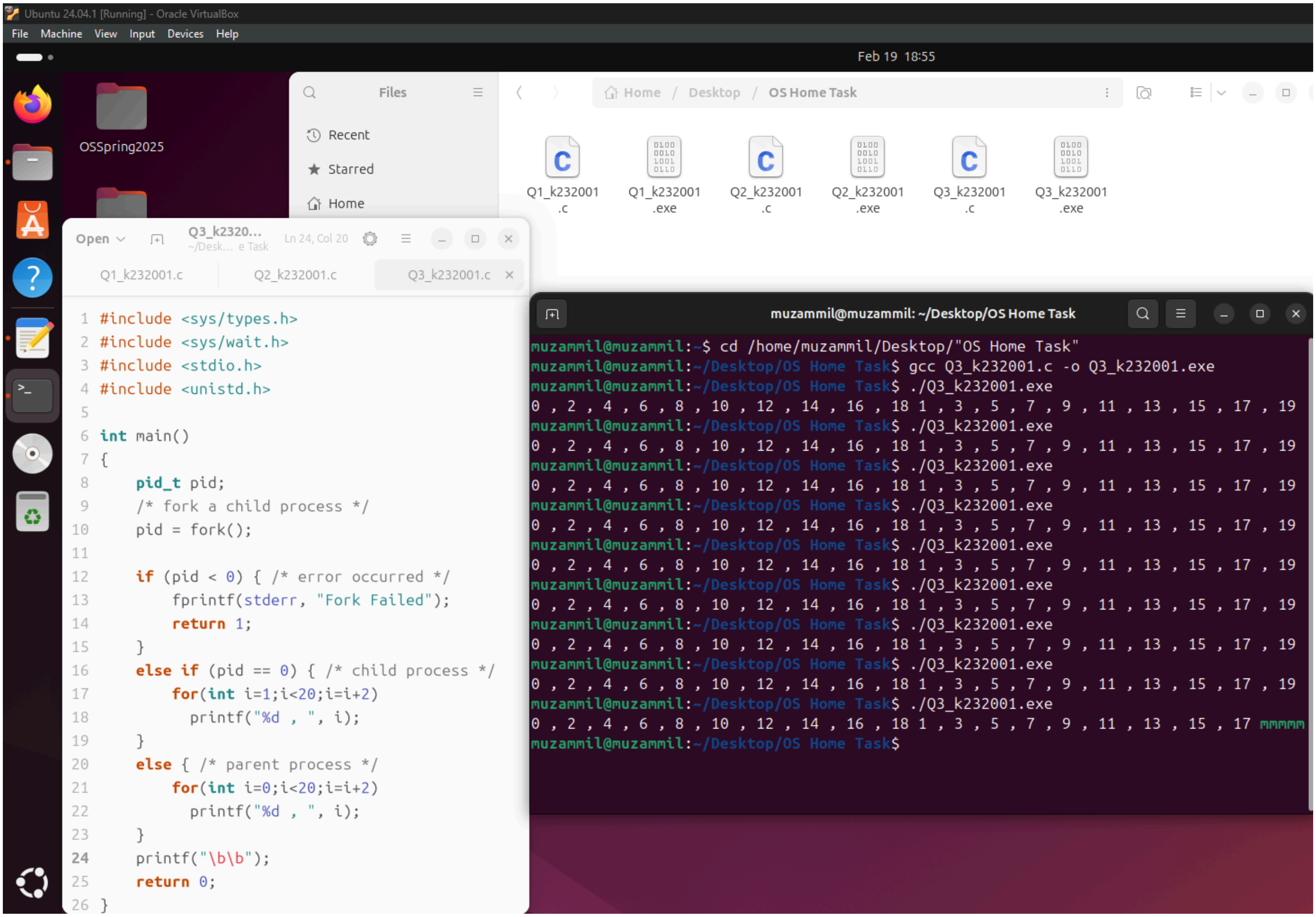
```
muzammil@muzammil:~/Desktop/OS Home Task$ gedit Q2_k232001.c
muzammil@muzammil:~/Desktop/OS Home Task$ gcc Q2_k232001.c -o Q2_k232001.exe
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q2_k232001.exe
[p_id: 4257 | parent_id: 4256] This is from the child process.
[p_id: 4258 | parent_id: 4257] This is from the grandchild process.
[p_id: 4257] Grandchild Complete
[p_id: 4256] Child Complete
muzammil@muzammil:~/Desktop/OS Home Task$
```



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

int main()
{
    pid_t pid1;
    /* fork a child process */
    pid1 = fork();
    if (pid1 < 0) { /* error occurred */
        fprintf(stderr, "Fork # 1 Failed");
        return 1;
    }
    else if (pid1 == 0) { /* child process */
        printf("[p_id: %d | parent_id: %d] This is from the child process.\n", getpid(), getppid());
        pid_t pid2 = fork();
        if (pid2 < 0) { /* error occurred */
            fprintf(stderr, "Fork # 2 Failed");
            return 1;
        }
        else if (pid2 == 0) { /* grandchild process */
            printf("[p_id: %d | parent_id: %d] This is from the grandchild process.\n", getpid(), getppid());
        }
        else { /* parent process */
            /* parent will wait for the child to complete */
            wait(NULL);
            printf("[p_id: %d] Grandchild Complete\n", getpid());
        }
    }

    else { /* grandparent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("[p_id: %d] Child Complete\n", getpid());
    }
    return 0;
}
```

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

int main()
{
	pid_t pid1;
	/* fork a child process */
	pid1 = fork();
	if (pid1 < 0) { /* error occurred */
	fprintf(stderr, "Fork # 1 Failed");
	return 1;
	}
	else if (pid1 == 0) { /* child process */
	printf("[p_id: %d | parent_id: %d] This is from the child process.\n", getpid(), getppid());
	pid_t pid2 = fork();
	if (pid2 < 0) { /* error occurred */
	fprintf(stderr, "Fork # 2 Failed");
	return 1;
	}
	else if (pid2 == 0) { /* grandchild process */
	printf("[p_id: %d | parent_id: %d] This is from the grandchild process.\n", getpid(),
getppid());
	}
	else { /* parent process */
	/* parent will wait for the child to complete */
	wait(NULL);
	printf("[p_id: %d] Grandchild Complete\n", getpid());
	}
	}

	else { /* grandparent process */
	/* parent will wait for the child to complete */
	wait(NULL);
	printf("[p_id: %d] Child Complete\n", getpid());
	}
	return 0;
}
```

Q3:



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

int main()
{
    pid_t pid;
    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        for(int i=1;i<20;i=i+2)
          printf("%d , ", i);
    }
    else { /* parent process */
        for(int i=0;i<20;i=i+2)
          printf("%d , ", i);
    }
    printf("\b\b");
    return 0;
}
```

```
muzammil@muzammil:~$ cd /home/muzammil/Desktop/"OS Home Task"
muzammil@muzammil:~/Desktop/OS Home Task$ gcc Q3_k232001.c -o Q3_k232001.exe
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 , 19
muzammil@muzammil:~/Desktop/OS Home Task$ ./Q3_k232001.exe
0 , 2 , 4 , 6 , 8 , 10 , 12 , 14 , 16 , 18 1 , 3 , 5 , 7 , 9 , 11 , 13 , 15 , 17 mmmmm
muzammil@muzammil:~/Desktop/OS Home Task$
```

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

int main()
{
        pid_t pid;
        /* fork a child process */
        pid = fork();

        if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
        }
        else if (pid == 0) { /* child process */
        for(int i=1;i<20;i=i+2)
        printf("%d , ", i);
        }
        else { /* parent process */
        for(int i=0;i<20;i=i+2)
        printf("%d , ", i);
        }
        printf("\b\b");
        return 0;
}
```