Laboratory Assignments 4

Subject: Design Principles of Operating Systems

Subject code: CSE 3249

Assignment 4: Familiarization with Process Management in Linux environment.

Objective of this Assignment:

- To trace the different states of a process during its execution.
- To learn the use of different system calls such as (fork(),vfork(),wait(),exec()) for process handling in Unix/Linux environment.

1. Write a C program to create a child process using fork() system call. The child process will print the message “Child” with its process identifier and then continue in an indefinite loop. The parent process will print the message “Parent” with its process identifier and then continue in an indefinite loop. a) Run the program and trace the state of both processes. b) Terminate the child process. Then trace the state of processes. c) Run the program and trace the state of both processes. Terminate the parent process. Then trace the state of processes. d) Modify the program so that the parent process after displaying the message will wait for child process to complete its task. Again run the program and trace the state of both processes. e) Terminate the child process. Then trace the state of processes.

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q1.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int  main() {
     pid_t pid = fork();

     if (pid < 0 ){
         perror("fork failed");
         return 1;
     }
     if (pid == 0){
         printf("Child Process | PID = %d\n" , getpid());
         while (1); // infinite loop
     }
     else {
         printf("Parent Process | PID = %d\n" , getpid());
         while(1); // infinite loop
     }
     return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q1.c -o a4q1
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q1
Parent Process | PID = 633
Child Process | PID = 634
```

2. Trace the output of the following codes:

a)
```
int main( )
{
   if(fork()==0)
      printf("1");
   else
      printf("2");
   printf("3");
return 0;
}
```

b)
```
int main( )
{
   if(vfork()==0)
   {
         printf("1");
         _exit(0);

   }
   else
     printf("2");
     printf("3");
 }
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2a.c
#include <stdio.h>
#include <unistd.h>

int main(){
    if(fork() == 0){
       printf("1");
    }
    else{
       printf("2");
    }
    printf("3");
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2a.c -o a4q2a
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2a
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2a
2313koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2b.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
   if(vfork() == 0)
   {
      printf("1");
      exit(0);
   }
   else
      printf("2");


   printf("3");
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2b.c -o a4q2b
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2b
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2b
123koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$
```

c)

```
int main( )
{
 pid_t pid;
 int i=5;
 pid=fork();
 i=i+1;
 if(pid= =0)
 {
     printf("Child: %d",i);
 }
 else

 {
     wait(NULL);
     printf("Parent: %d",i);
 }
 return 0;
}
```

d)

```
int main( )
{
 pid_t pid;
 int i=5;
 pid=vfork();
 i=i+1;
 if(pid==0)
 {
     printf("Child: %d",i);
     _exit(0);
 }

 else
 {
     printf("Parent: %d",i);
 }
 return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat a4q2c.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
int main(){
    pid_t pid;
    int i = 5;
    pid = fork();
    i = i + 1 ;
    if(pid == 0){
      printf("Child: %d\n" , i);
    }
    else {
      wait(NULL);
      printf("Parent: %d\n" , i);
    }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2c.c -o a4q2c
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2c
Child: 6
Parent: 6
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2d.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
int main() {
    pid_t pid;
    int i = 5;

    pid = vfork();
    i = i + 1;

    if (pid == 0) {
        printf("Child: %d\n", i);
        _exit(0);   // SAFER for vfork()
    }
    else {
        printf("Parent: %d\n", i);
    }

    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2d.c -o a4q2d
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2d
Child: 6
Parent: 7
```

e)

```
int main( )
{
 pid_t pid;
 int i=5;
 pid=fork();
 if(pid= =0)
 {
    i=i+1;
    printf("Child: %d",i);
 }
  else
 {
  wait(NULL);
  printf("Parent: %d",i);
 }
 return 0;
}
```

f)

```
int main( )
{
  pid_t pid;
  int i=5;
   pid=vfork();
   if(pid==0)
   {
      i=i+1;
      printf("Child: %d",i);
      _exit(0);
   }
   else
   {
      printf("Parent: %d",i);
   }
   return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2e.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
int main(){
    pid_t pid;
    int i = 5 ;
    pid = fork();
     if(pid == 0 ){
       i = i + 1 ;
       printf("Child: %d\n", i);
    }
    else {
        wait(NULL);
        printf("Parent: %d\n", i);
    }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2e.c -o a4q2e
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2e
Child: 6
Parent: 5
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2f.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(){
    pid_t pid;
    int i = 5;
    pid = vfork();
    if(pid == 0){
      i = i + 1 ;
      printf("Child: %d\n", i);
      _exit(0);
    }
    else {
       printf("Parent: %d\n" , i);
    }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2f.c -o a4q2f
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2f
Child: 6
Parent: 6
```

g)

```
int main( )
{
 int i=5;
 if(fork( )==0)
 {
     printf("Child: %d",i);
 }
 else
 {
    printf("Parent: %d",i);
 }
 return 0;
}
```

h)

```
int main( )
{
 int i=5;
 if(vfork( )==0)
 {
     printf("Child: %d",i);
     _exit(0);
 }
 else
 {
     printf("Parent: %d",i);
 }
  return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2g.c
#include <stdio.h>
#include <unistd.h>
int main(){
     int i = 5;
     if(fork() == 0){
        printf("Child: %d\n" , i);
     }
     else{
        printf("Parent: %d\n" , i);
     }
     return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2g.c -o a4q2g
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2g
Parent: 5
Child: 5
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2h.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    int i = 5;
    if(vfork() == 0){
       printf("Child: %d\n" , i);
       _exit(0);
    }
    else {
       printf("Parent: %d\n" , i);
    }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2h.c -o a4q2h
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2h
Child: 5
Parent: 5
```

i)
```
int main( )
{
 if(fork( )==0)
 {
     printf("1");
 }
 else
 {
     wait(NULL);
     printf("2");
     printf("3");
  }
  return 0;
 }
```

j)
```
int main( )
{
 if(vfork( )==0)
 {
     printf("1");
     _exit(0);
 }
  else
 {
     printf("2");
    printf("3");
  }
  return 0;
 }
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2i.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int main(){
    if(fork() == 0){
       printf("1");
    }
    else{
     wait(NULL);
     printf("2");
     printf("3");
   }
   return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2i.c -o a4q2i
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2i
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2i
123koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat a4q2j.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main() {
    if(vfork() == 0){
      printf("1");
      _exit(0);
    }
    else{
      printf("2");
      printf("3");
    }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2j.c -o a4q2j
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2j
123koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$
```

k)

```
int main( )
{
pid_t c1;
int n=10;
c1=fork( );
if(c1==0)
{
      printf(" Child\n");
      n=20;
      printf("n=%d \n",n);
 }
 else
 {
       wait(NULL);
       printf("Parent\n");
       printf("n=%d \n",n);
  }
   return 0;
}
```

l)

```
int main( )
{
 pid_t c1;
 int n=10;
 c1=vfork( );
 if(c1==0)
 {
       printf(" Child\n");
       n=20;
       printf("n=%d \n",n);
       _exit(0);
 }
 else
 {
       printf("Parent\n");
       printf("n=%d \n",n);
 }
  return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2k.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
int main(){
    pid_t c1;
    int n = 10;
    c1 = fork();
    if(c1 == 0) {
      printf("Child\n");
      n = 20;
      printf("n=%d\n", n);
    }
    else {
        wait(NULL);
        printf("Parent\n");
        printf("n=%d\n",n);
     }
     return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2k.c -o a4q2k
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2k
Child
n=20
Parent
n=10
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2l.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
int main() {
    pid_t c1;
    int n = 10;
    c1 = vfork();
    if(c1 == 0){
       printf("Child\n");
       n=20;
       printf("n=%d\n", n);
       _exit(0);
     }
     else{
       printf("Parent\n");
       printf("n=%d\n" , n);
     }
     return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2l.c -o a4q2l
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2l
Child
n=20
Parent
n=20
```

m)

```
int main( )
{
 int i=5;
 fork();
 i=i+1;
 fork();
 printf ( "% d",i);
 return 0;
}
```

n)

```
int main( )
{
 pid_t pid;
 int i=5;
 pid=vfork();
 if(pid==0)
 {
      printf("Child: %d",i);
      _exit(0);
 }
  else
  {
       i=i+1;
       printf("Parent: %d",i);
   }
   return 0;
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2m.c
#include <stdio.h>
#include <unistd.h>
int main(){
    int i = 5;
    fork();
    i = i + 1;
    fork();
    printf("%d\n" , i);
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2m.c -o a4q2m
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2m
6
6
6
6
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat a4q2n.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
int main(){
    pid_t pid;
    int i = 5;
    pid = vfork();
    if(pid == 0){
       printf("Child: %d\n" , i);
       _exit(0);
    }
    else {
       i = i + 1 ;
       printf("Parent: %d\n" , i);
    }
   return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2n.c -o a4q2n
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2n
Child: 5
Parent: 6
```

o)
```
int main( )
{
 int i=5;
 if(fork()==0)
 i=i+1;
 else
 i=i-1;
 printf("%d",i);
 return 0;
}
```

p)
```
int main( )
{
 int i=5;
 if(vfork()==0)
 {
      i=i+1;
      _exit(0);
  }
  else
       i=i-1;
  fprintf(stderr,"%d",i);
  return 0;
  }
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat a4q2o.c
#include <stdio.h>
#include <unistd.h>
int main(){
    int i = 5;
    if(fork() == 0){
      i = i + 1 ;
    }
    else{
      i = i - 1;
     }
    printf("%d\n", i);
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2o.c -o a4q2o
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2o
4
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ 6
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2p.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    int i = 5 ;
    if(vfork() == 0){
      i = i + 1;
      _exit(0);
    }
    else{
      i = i - 1 ;
    }
   fprintf(stderr , "%d\n" , i);
   return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2p.c -o a4q2p
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2p
5
```

q)

```
int main( )
{
int j,i=5;
for(j=1;j<3;j++)
{
    if(fork()==0)
    {
        i=i+1;
        break;
    }
else
   wait(NULL);
}
printf("%d",i);
return 0;
}
```

r)

```
int main( )
{
int j,i=5;
for(j=1;j<3;j++)
{
    if(fork()!=0)
    {
        i=i-1;
        break;
    }
}
fprintf(stderr,"%d",i);
return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2q.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int main() {
    int j, i = 5;
    for(j = 1 ; j < 3 ; j++){
       if(fork() == 0){
          i = i + 1 ;
          break;
       }
       else{
          wait(NULL);
       }
    }
    printf("%d\n" , i);
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2q.c -o a4q2q
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2q
6
6
5
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2r.c
#include <stdio.h>
#include <unistd.h>
int main(){
     int j, i = 5;
     for(j=1 ; j<3 ; j++){
         if(fork() != 0){
          i = i - 1 ;
            break;
          }
      }
     fprintf(stderr, "%d\n" , i);
     return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2r.c -o a4q2r
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2r
4
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ 4
5
```

s)

```
int main( )
{
if(fork() == 0)
if(fork())
printf("1\n");

return 0;
}
```

t)

```
void fun1(){
fork();
fork();
printf("1\n");
}


int main() {
fun1();
printf("1\n");
return 0;
}
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2s.c
#include <stdio.h>
#include <unistd.h>
int main(){
    if(fork() == 0){
       if(fork()){
          printf("1\n");
       }
    }
   return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2s.c -o a4q2s
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2s
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ 1
```

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q2t.c
#include <stdio.h>
#include <unistd.h>
void fun1() {
     fork();
     fork();
     printf("1\n");
}
int main(){
    fun1();
    printf("1\n");
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q2t.c -o a4q2t
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q2t
1
1
1
1
1
1
1
1
```

3. Write a C program that will create three child process to perform the following operations respectively: - - - - - First child will copy the content of file1 to file2 Second child will display the content of file2 Third child will display the sorted content of file2 in reverse order. Each child process being created will display its id and its parent process id with appropriate message. The parent process will be delayed for 1 second after creation of each child process. It will display appropriate message with its id after completion of all the child processes.

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q3.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>

int main() {
    pid_t pid;
    FILE *f1, *f2;
    char ch;

    // ------------------ FIRST CHILD ------------------
    pid = fork();
    if (pid == 0) {
        printf("First Child Created | PID = %d | Parent PID = %d\n", getpid(), getppid());

        f1 = fopen("file1.txt", "r");
        f2 = fopen("file2.txt", "w");

        if (f1 == NULL || f2 == NULL) {
            printf("File Error\n");
            _exit(1);
        }

        while ((ch = fgetc(f1)) != EOF)
            fputc(ch, f2);

        fclose(f1);
        fclose(f2);
        _exit(0);
```

```
    }

    sleep(1);
    wait(NULL);

    // ------------------ SECOND CHILD ------------------
    pid = fork();
    if (pid == 0) {
        printf("Second Child Created | PID = %d | Parent PID = %d\n", getpid(), getppid());

        f2 = fopen("file2.txt", "r");

        if (f2 == NULL) {
            printf("File Error\n");
            _exit(1);
        }

        printf("\nContents of file2:\n");
        while ((ch = fgetc(f2)) != EOF)
            putchar(ch);

        fclose(f2);
        _exit(0);
    }

    sleep(1);
    wait(NULL);

    // ------------------ THIRD CHILD ------------------
    pid = fork();
    if (pid == 0) {
```

```
    // ------------------ THIRD CHILD ------------------
    pid = fork();
    if (pid == 0) {
        printf("\nThird Child Created | PID = %d | Parent PID = %d\n", getpid(), getppid());
        system("sort -r file2.txt");
        _exit(0);
    }

    sleep(1);
    wait(NULL);

    // ------------------ PARENT ------------------
    printf("\nParent Process Completed | PID = %d\n", getpid());
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q3.c -o a4q3
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q3
First Child Created | PID = 612 | Parent PID = 611
File Error
Second Child Created | PID = 613 | Parent PID = 611

Contents of file2:

Third Child Created | PID = 614 | Parent PID = 611

Parent Process Completed | PID = 611
```

4. Write a C program that will create a child process to generate a Fibonacci series of specified length and store it in an array. The parent process will wait for the child to complete its task and then display the Fibonacci series and then display the prime Fibonacci number in the series along with its position with appropriate message.

```
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ cat > a4q4.c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int isPrime(int n){
    if(n<2) return 0;
    for(int i = 2 ; i*i <= n; i++) if(n%i == 0) return 0;
    return 1;
}
int main() {
    int n ;
    printf("Enter length : ");
    scanf("%d" , &n);
    int arr[n];
    pipe((int*)arr);
    if(fork() == 0){
       int a=0,b=1,c;
       for(int i = 0 ; i<n ;i++){
           arr[i]=a;
           c=a+b;
           a=b;
           b=c;
        }
        _exit(0);
   }
  else {
       wait(NULL);
       printf("Fibonacci Series:\n");
       for(int i=0 ; i<n ; i++){
          printf("%d " , arr[i]);
          if(isPrime(arr[i]))
                printf(" ( Prime at position %d)\n" , i+1 );
          }
     }
    return 0;
}
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ gcc a4q4.c -o a4q4
koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$ ./a4q4
Enter length : 10
Fibonacci Series:
3  ( Prime at position 1)
4 0 0 0 0 0 0 0 0 koushik_das@LAPTOP-BIHGEI3G:~/DOS_2341004117/DOSass4$
```