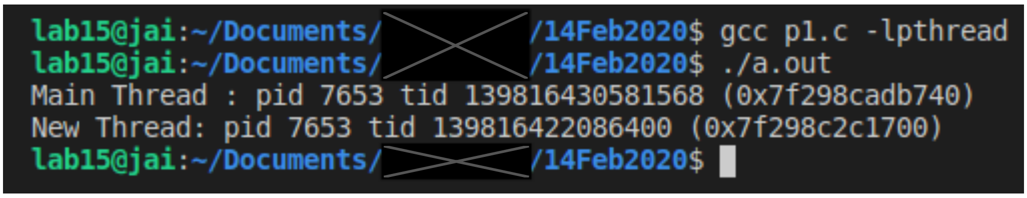
Program 1 :

```
#include<stdio.h>
#include<unistd.h>
#include<pthread.h>
#include<stdlib.h>

pthread_t ntid;
void printids(const char*s)
{
pid_t pid;
pthread_t tid;
pid = getpid();
tid = pthread_self();
//printf("%s pid %u tid %u (0x%x)\n", s, (unsigned int)pid, (unsigned int)tid, (unsigned int)tid);
printf("%s pid %u tid %lu (0x%lx)\n", s, pid, tid, tid);
}

void *thr_fn(void *arg)
{
printids("New Thread:");
return((void*)0);
}

void main()
{
int err;
err = pthread_create(&ntid, NULL, thr_fn, NULL);
if(err!=0)
{
printf("Error");
}
printids("Main Thread :");
sleep(1);
exit(0);
}
```



```
lab15@jai:~/Documents/[redacted]/14Feb2020$ gcc p1.c -lpthread
lab15@jai:~/Documents/[redacted]/14Feb2020$ ./a.out
Main Thread : pid 7653 tid 139816430581568 (0x7f298cadb740)
New Thread: pid 7653 tid 139816422086400 (0x7f298c2c1700)
lab15@jai:~/Documents/[redacted]/14Feb2020$ █
```

Program 2 :

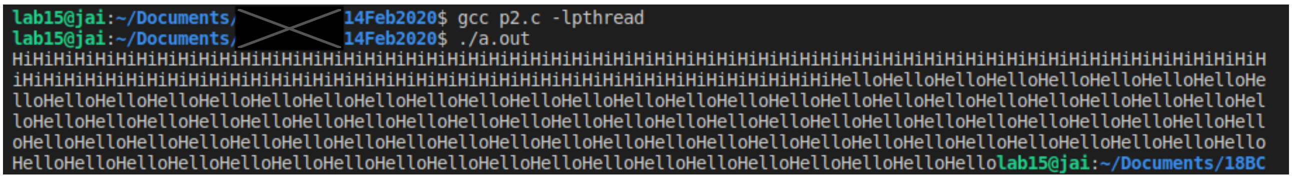
```
#include<stdio.h>
#include<unistd.h>
#include<pthread.h>
#include<stdlib.h>

pthread_t ntid, ntid1;

void *thr_fn(void *arg)
{
for(int i=0;i<100;i++)
{
printf("Hi");
}
return((void*)0);
}

void *thr_fn1(void *arg)
{
for(int i=0;i<100;i++)
{
printf("Hello");
}
return((void*)0);
}

void main()
{
int err, err1;
err = pthread_create(&ntid, NULL, thr_fn, NULL);
err1 = pthread_create(&ntid1, NULL, thr_fn1, NULL);
if(err!=0 && err1!=0)
{
printf("Error");
}
pthread_join(ntid, NULL);
pthread_join(ntid1, NULL);
sleep(1);
exit(0);
}
```



```
lab15@jai:~/Documents/[redacted]14Feb2020$ gcc p2.c -lpthread
lab15@jai:~/Documents/[redacted]14Feb2020$ ./a.out
HiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiH
iHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHiHelloHelloHelloHelloHelloHelloHelloHelloHe
lloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHel
loHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHell
oHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHello
HelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHellolab15@jai:~/Documents/18BC
```

Program 3 :

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>

sem_t mutex;
pthread_t ntid, ntid1;

void *hi(void *arg)
{
sem_wait(&mutex);
int i;
for(i=0;i<50;i++)
{
printf("Hi");
}
printf("\n");
sem_post(&mutex);
}

void *hello(void *arg)
{
sem_wait(&mutex);
int i;
for(i=0;i<50;i++)
{
printf("Hello");
}
printf("\n");
sem_post(&mutex);
}

void main()
{
int err, err1;
err = pthread_create(&mutex, NULL, hi, NULL);
err1 = pthread_create(&mutex, NULL, hello, NULL);
if(err!=0 )
{
printf("Error");
}
pthread_join(ntid, NULL);
pthread_join(ntid1, NULL);
sleep(1);
exit(0);
}
```

Program 4 :

```
#include<stdio.h>
#include<unistd.h>
#include<pthread.h>
#include<stdlib.h>

pthread_t ntid, ntid1;


void *thr_fn(void *arg)
{
   int n=10, first = 0, second = 1, next, c;
   printf("Fibonacci series of 10 terms : ");
   for (c = 0; c < n; c++)
   {
      if (c <= 1)
      next = c;
      else
      {
      next = first + second;
      first = second;
      second = next;
      }
      printf("%d ", next);
   }
   printf("\n");
   return((void*)0);
}

void *thr_fn1(void *arg)
{
   printf("Sorting : \n");
   int n=5, c, d, swap;
   int array[5] = {1, 3, 2, 5, 4};

   for (c = 0 ; c < n - 1; c++)
   {
      for (d = 0 ; d < n - c - 1; d++)
      {
         if (array[d] > array[d+1])
         {
            swap       = array[d];
            array[d]   = array[d+1];
            array[d+1] = swap;
         }
      }
```

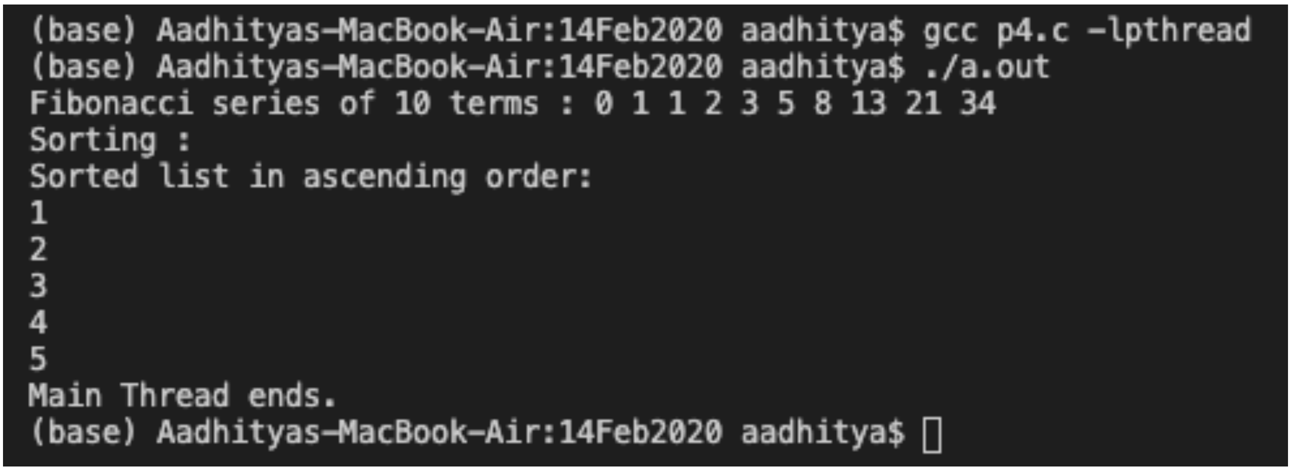
```
    }

    printf("Sorted list in ascending order:\n");

    for (c = 0; c < n; c++)
        printf("%d\n", array[c]);

    return((void*)0);
}

int main()
{
    int err, err1;
    err = pthread_create(&ntid, NULL, thr_fn, NULL);
    err1 = pthread_create(&ntid1, NULL, thr_fn1, NULL);
    if(err!=0 && err1!=0)
    {
        printf("Error");
    }
    pthread_join(ntid, NULL);
    pthread_join(ntid1, NULL);
    printf("Main Thread ends.\n");
    sleep(1);
    exit(0);
}
```



```
(base) Aadhityas-MacBook-Air:14Feb2020 aadhitya$ gcc p4.c -lpthread
(base) Aadhityas-MacBook-Air:14Feb2020 aadhitya$ ./a.out
Fibonacci series of 10 terms : 0 1 1 2 3 5 8 13 21 34
Sorting :
Sorted list in ascending order:
1
2
3
4
5
Main Thread ends.
(base) Aadhityas-MacBook-Air:14Feb2020 aadhitya$ ▯
```