1. **Write a C program to define 3 different threads with the following purposes where N is the input  
   ● Thread A - To run a loop and return the sum of first N prime numbers  
   ● Thread B & C - should run in parallel. One prints "Thread 1 running" every 2 seconds, and the other prints "Thread 2 running" every 3 seconds for 100 seconds.**

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

#include <time.h>

int is\_prime(int num) {

if (num < 2) return 0;

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) return 0;

}

return 1;

}

void\* prime\_sum(void\* arg) {

int n = \*((int\*)arg);

int count = 0, num = 2, sum = 0;

while (count < n) {

if (is\_prime(num)) {

sum += num;

count++;

}

num++;

}

printf("Sum of first %d prime numbers: %d\n", n, sum);

return NULL;

}

void\* thread1(void\* arg) {

time\_t start = time(NULL);

while (time(NULL) - start < 100) {

printf("Thread 1 running\n");

sleep(2);

}

return NULL;

}

void\* thread2(void\* arg) {

time\_t start = time(NULL);

while (time(NULL) - start < 100) {

printf("Thread 2 running\n");

sleep(3);

}

return NULL;

}

int main() {

pthread\_t t1, t2, t3;

int n;

printf("Enter N: ");

scanf("%d", &n);

pthread\_create(&t1, NULL, prime\_sum, &n);

pthread\_create(&t2, NULL, thread1, NULL);

pthread\_create(&t3, NULL, thread2, NULL);

pthread\_join(t1, NULL);

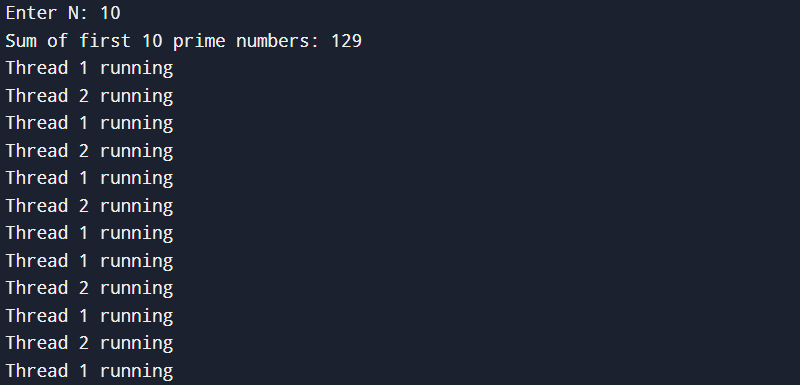
pthread\_join(t2, NULL);

pthread\_join(t3, NULL);

return 0;

}

**OUTPUT:**

****