1. **Write a multi-threaded program that generates the Fibonacci series p-threads / Win32 thread libraries, this program should work as follows, the number of Fibonacci numbers that the program is to generate, the program will then create a separate thread that will generate the fibonacci numbers placing the sequence in data that is shared by the threads. When the thread finishes execution the parent thread will output the sequence generated by the child thread.**

**#include<stdio.h>  
#include<pthread.h>  
  
  
#define SIZE 25  
  
int fib\_arr[SIZE]; //this array is shared by threads  
void \*fibonachi(void \*n); // dymanically aloocate memory  
  
int main(int argc,char \*argcv[] ){  
  
 int num;  
 int x;  
  
 pthread\_attr\_t attr; //declare pthread\_t type variable 'tid'  
  
 if(argc !=2){  
 printf("Please insert one command line argument\n");  
 return -1;  
 }  
 num=atoi(argcv[1]);  //convert string value to integer and store to 'no'  
  
 if(num<0){  
 printf("To Countinue please enter non-negative number\n");  
 }  
 else{  
 pthread\_attr\_init(&attr);  
 pthread\_t tid; //thread Identifier  
 pthread\_create(&tid,&attr,fibonachi,num); //create a thread  
 pthread\_join(tid,NULL);  
  
 printf(" - Programs Out put - \n");  
 printf("\n");  
 printf("Set of Fibonachi numbers.........\n");  
  
 for(x=0; x<num; x++){  
  
 printf("%d ",fib\_arr[x]);  
 }  
 printf("\n");  
   
 }  
 return 0;  
}  
  
void \*fibonachi(void \*n){  
  
 int x = 0;  
 int y = 1;  
 int fibn=0;  
 int i;  
   
 for(i=1; i<=n; i++){  
 fib\_arr[i-1]=fibn;  
 fibn = x + y;  
 x = y;  
 y = fibn;  
 }   
   
 pthread\_exit(0);  
}**

1. **Write a c program using the fork system call that generates the Fibonacci sequence in the child process the number of the sequence will be provided in the command line.**

**#include <stdio.h>**

**#include <sys/types.h>**

**#include <unistd.h>**

**int main()**

**{**

**int a=0, b=1, n=a+b,i,ii;**

**pid\_t pid;**

**printf("Enter the number of a Fibonacci Sequence:\n");**

**scanf("%d", &ii);**

**if (ii < 0)**

**printf("Please enter a non-negative integer!\n");**

**else**

**{**

**pid = fork();**

**if (pid == 0)**

**{**

**printf("Child is producing the Fibonacci Sequence...\n");**

**printf("%d %d",a,b);**

**for (i=0;i<ii;i++)**

**{**

**n=a+b;**

**printf("%d ", n);**

**a=b;**

**b=n;**

**}**

**printf("Child ends\n");**

**}**

**else**

**{**

**printf("Parent is waiting for child to complete...\n");**

**wait(NULL);**

**printf("Parent ends\n");**

**}**

**}**

**return 0;**

**}**