// Que 1

```
#include<pthread.h>
#include<stdio.h>
int goal=0;
long fib[500];
void *runner(void *param)
{
      if(goal<=0)
            pthread_exit(0);
      fib[0]=0;
      if(goal>1)
      {
            fib[1]=1;
            for(int i=2;i<goal;i++)
                  fib[i]=fib[i-1]+fib[i-2];
            pthread_exit(0);
      }
}
int main(int argc, char *argv[])
{
      pthread_t tid;
      pthread_attr_t attr;
      pthread_attr_init(&attr);
      printf("Print this many Fibonacci numbers: ");
      scanf("%d", &goal);
      if(goal>500)
      {
            printf("Printing as many as possible: 500\n");
            goal = 500;
      }
      pthread_create(&tid,&attr,runner,argv[1]);
      pthread_join(tid, NULL);
      if(goal>0)
      {
            printf("%ld", fib[0]);
      }
      for (int i=1;i<goal;i++)
      {
            printf(", %ld", fib[i]);
      }
      printf("\n");
      return 0;
}
```

//Que 2.

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


int glob = 0;
void *myThreadFun(void *vargp)
{
    int *myid = (int *)vargp;
```

```
    int locl =1;

    static int st = 0;

    ++st; ++glob;
    printf("Thread ID: %d, Static: %d, Global: %d, Local: %d\n", *myid, ++st, +
+glob,++locl);
}

int main()
{
    int i;
    pthread_t tid;

    for (i = 0; i < 3; i++)
        pthread_create(&tid, NULL, myThreadFun, (void *)&tid);

    pthread_exit(NULL);
    return 0;
}
```