|  |  |  |  |
| --- | --- | --- | --- |
| **Lab No** | 07 | **Reg. No** | 224921 |
| **Student Name** | Muhammad Saad Tariq | **Section** | A |

|  |
| --- |
| Solution Task 1 |
| Code:  #include <pthread.h>  #include <stdio.h>  #define FIRST\_ODD\_NUM 1  #define FIRST\_EVEN\_NUM 2  #define MAX 10  void \*print\_even\_nums(void \*param) {  int even\_num\_to\_print;  for(even\_num\_to\_print = FIRST\_EVEN\_NUM; even\_num\_to\_print < MAX;) {  printf("%d ", even\_num\_to\_print);  even\_num\_to\_print += 2;  }  }  void \*print\_odd\_nums(void \*param) {  int odd\_num\_to\_print;  for(odd\_num\_to\_print = FIRST\_ODD\_NUM; odd\_num\_to\_print < MAX;) {  printf("%d ", odd\_num\_to\_print);  odd\_num\_to\_print += 2;  }  }  int main(int argc, char \*argv[]) {  pthread\_t even\_thread;  pthread\_t odd\_thread;  pthread\_create(&even\_thread, NULL, print\_even\_nums, NULL);  pthread\_create(&odd\_thread, NULL, print\_odd\_nums, NULL);  pthread\_join(even\_thread, NULL); pthread\_join(odd\_thread, NULL);  printf("\n");  }  Output: |

|  |
| --- |
| Solution Task 2 |
| Code:  #include <pthread.h>  #include <stdio.h>  #include <semaphore.h>  #define FIRST\_ODD\_NUM 1  #define FIRST\_EVEN\_NUM 2  #define MAX 10  sem\_t sem1;  sem\_t sem2;  void \*print\_even\_nums(void \*param) {  int even\_num\_to\_print;  for(even\_num\_to\_print = FIRST\_EVEN\_NUM; even\_num\_to\_print < MAX;) {  sem\_wait(&sem1);  printf("%d ", even\_num\_to\_print);  even\_num\_to\_print += 2;  sem\_post(&sem2);  }  }  void \*print\_odd\_nums(void \*param) {  int odd\_num\_to\_print;  for(odd\_num\_to\_print = FIRST\_ODD\_NUM; odd\_num\_to\_print < MAX;) {  sem\_wait(&sem2);  printf("%d ", odd\_num\_to\_print);  odd\_num\_to\_print += 2;  sem\_post(&sem1);  }  }  int main(int argc, char \*argv[]) {  pthread\_t even\_thread;  pthread\_t odd\_thread;  sem\_init(&sem1, 0, 0);  sem\_init(&sem2, 0, 1);  pthread\_create(&even\_thread, NULL, print\_even\_nums, NULL);  pthread\_create(&odd\_thread, NULL, print\_odd\_nums, NULL);  pthread\_join(even\_thread, NULL); pthread\_join(odd\_thread, NULL);  printf("\n");  }  Output: |