OS Lab 8

Aniruddha Amit Dutta

180905488

Roll – 58

batch 9

Q1.

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

struct s_strct{
        int num;
        int arr[100];
};


void* thread_code(void *s){
        struct s_strct *s2 = (struct s_strct*)s;
        int a = (int)s2->num;
        int* arr = (int *)s2->arr;
        printf("in thread_code \n");

        arr[0]=0;arr[1]=1;
        for(int i=2;i<a;++i){
                arr[i] = arr[i-1]+arr[i-2];
        }
}

int main()
{
        pthread_t tid;
        int n;
        struct s_strct * s;
        s = malloc(sizeof(struct s_strct));
        printf("enter num\n");
```

```
        scanf("%d",&n);
        s->num = n;
        pthread_create(&tid,0,&thread_code,(void *)s);
        pthread_join(tid,0); // wait for hread to complete
        printf("in main \n");
        printf("fibbonaci series is\n");
        for (int i = 0; i < n; ++i)
        {
                printf("%d ",s->arr[i] );
        }

}
```

```
$ gcc fib.c -lpthread
$ ./a.out
enter num
6
in thread_code
in main
fibbonaci series is
0 1 1 2 3 5 $ ./a,out
sh: 98: ./a,out: not found
$ ./a.out
enter num
23
in thread_code
in main
fibbonaci series is
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711
 $
```

Q2.

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

struct s_strct{
        int num;
        int arr[100];
};


void* thread_code(void *s){
        struct s_strct *s2 = (struct s_strct*)s;
        int a = (int)s2->num;
```

```
        int* arr = (int *)s2->arr;
        printf("in thread_code \n");

        int sum = 0;
        for(int i=0;i<a;++i){
                sum += arr[i];
        }
        return (void *)sum;
}

int main()
{
        pthread_t tid;
        int n,result;
        struct s_strct * s;
        s = malloc(sizeof(struct s_strct));
        printf("enter tot num \n");
        scanf("%d",&n);
        s->num = n;
        for (int i = 0; i < n; ++i)
        {
                scanf("%d",&(s->arr[i]));
        }
        pthread_create(&tid,0,&thread_code,(void *)s);
        pthread_join(tid,(void**)&result ); // wait for hread to complete
        printf("in main \n");
        printf("sum is = %d\n",result );


}
```

output -

```
aniruddha@aniruddha-G3-3579:~/Desktop$ gcc sum.c -lpthread
sum.c: In function ‘thread_code’:
sum.c:23:9: warning: cast to pointer from integer of different size [-Wint-to-po
inter-cast]
  return (void *)sum;
         ^
aniruddha@aniruddha-G3-3579:~/Desktop$ ./a.out
enter tot num
5
1 2 3 4 5
in thread_code
in main
sum is = 15
aniruddha@aniruddha-G3-3579:~/Desktop$
```

Q3.

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

struct s_strct{
        int l;
        int u;
        int arr[100];
};


void* thread_code(void *s){
        struct s_strct *s2 = (struct s_strct*)s;
        int l = (int)s2->l;
        int u = (int)s2->u;
        int* arr = (int *)s2->arr;
        printf("in thread_code \n");

        int i,j,c=0,k=1;
        for(i=l;i<=u;i++)
        {
           c=0;
                for(j=1;j<=i;j++)
           {
              if(i%j==0)
              {
                 c++;
              }
           }
           if(c==2)
           {
              arr[k++]=i;
           }
        }
        arr[0]=k;

}

int main()
{
```

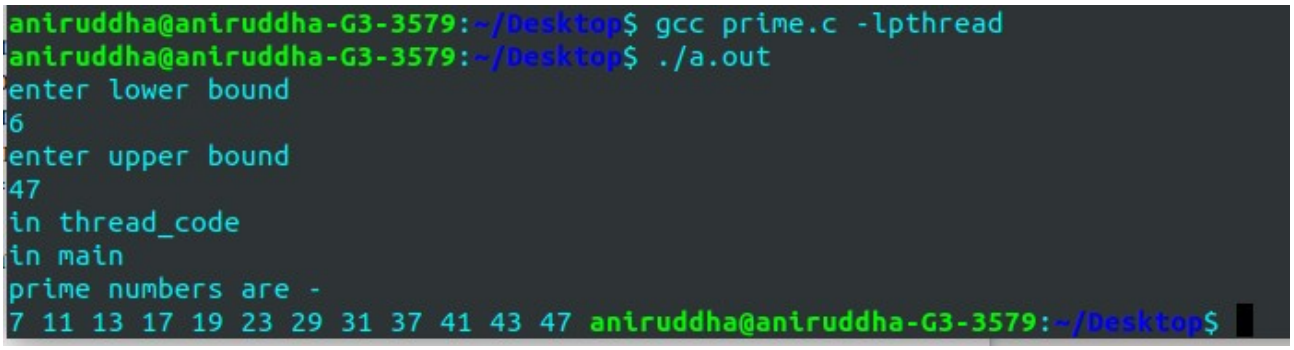
```
	pthread_t tid;
	int n,m;
	struct s_strct * s;
	s = malloc(sizeof(struct s_strct));
	printf("enter lower bound \n");
	scanf("%d",&n);
	printf("enter upper bound \n");
	scanf("%d",&m);

	s->l = n;
	s->u = m;

	pthread_create(&tid,0,&thread_code,(void *)s);
	pthread_join(tid,0); // wait for hread to complete
	printf("in main \n");
	printf("prime numbers are -\n");
	for (int i = 1; i < s->arr[0]; ++i)
	{
		printf("%d ",s->arr[i] );
	}

}
```

output -



Q4.

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

struct s_strct{
	int num;
```

```
        int sumeven;
        int sumodd;
        int arr[100];
};


void* thread_code1(void *s){
        struct s_strct *s2 = (struct s_strct*)s;
        int a = (int)s2->num;
        int* arr = (int *)s2->arr;
        printf("in thread_code1 \n");

        s2->sumeven = 0;
        for(int i=0;i<a;++i){
                if(arr[i]%2==0){
                        s2->sumeven += arr[i];
                }
        }
}

void* thread_code2(void *s){
        struct s_strct *s2 = (struct s_strct*)s;
        int a = (int)s2->num;
        int* arr = (int *)s2->arr;
        printf("in thread_code2 \n");

        s2->sumodd = 0;
        for(int i=0;i<a;++i){
                if(arr[i]%2==1){
                        s2->sumodd += arr[i];
                }
        }
}

int main()
{
        pthread_t tid1,tid2;
        int n;
        struct s_strct * s;
        s = malloc(sizeof(struct s_strct));
        printf("enter tot num \n");
        scanf("%d",&n);
        s->num = n;
        printf("enter nums \n");
        for (int i = 0; i < n; ++i)
```

```
        {
                scanf("%d",&(s->arr[i]));
        }

        pthread_create(&tid1,0,&thread_code1,(void *)s);
        pthread_create(&tid2,0,&thread_code2,(void *)s);

        pthread_join(tid1,0); // wait for thread to complete
        pthread_join(tid2,0);

        printf("in main \n");
        printf("sum odd is = %d\n",s->sumodd );
        printf("sum even is = %d\n",s->sumeven );

}
```

output -