Experiment 6:

Code 1:

```c
//SIMPLE first code of multi threading

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

void* abc(void* args);

int main()
{
    pthread_t t1;
    char *m = "5";
    pthread_create(&t1,NULL,abc,m);
    pthread_join(t1,NULL);


    //for(int i=20;i<26;i++)
    //{
    //  printf("%d ",i);
    //}

    return 0;
}




void* abc(void* args)
{

    int n = atoi(args);

    printf("Hello %d \n",n);
    for(int i=1;i<6;i++)
    {
        printf("%d ",i);
    }
}
```

Output 1:

```
Hello 5
1 2 3 4 5
```

Code 2:

```c
//FACT and digit sum

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

void* fact(void* args);
void* digisum(void* args);

int main()
{
    pthread_t t1;
    pthread_t t2;

    char* num = "5";
    pthread_create(&t1,NULL,fact,num);
    pthread_join(t1,NULL);

    char* num2 = "98458";
    pthread_create(&t2,NULL,digisum,num2);
    pthread_join(t2,NULL);

}

void* fact(void* args)
    {
        int n = atoi(args);
        int fac = 1;
        for(int i=1;i<=n;i++)
        {
            fac = fac*i;
        }
        printf("Factorial of %d is %d\n",n,fac);
    }

void* digisum(void* args)
    {
    int sum = 0;
        int m = atoi(args);
        int n = m;
```

```
        for(int i=1;i<=100;i++)
        {
            sum = sum + n%10;
            n=n/10;
            if(n==0)
            {
                printf("Digit sum of %d is %d\n",m,sum);
                exit(0);
            }
        }

    }
```

Output 2:

```
Factorial of 5 is 120
Digit sum of 98458 is 34
```

Code 3:

```
//RACE (AROUND) CONDITION

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

int balance = 100;

void* withdraw(void* args);
void* deposit(void* args);


int main()
{
    pthread_t t1;
    pthread_t t2;

    pthread_create(&t1,NULL,withdraw,NULL);
    pthread_create(&t2,NULL,deposit,NULL);

    pthread_join(t1,NULL);
    pthread_join(t2,NULL);//now t2 will run seprately from any other thread
```

```
}

void* withdraw(void* args)
    {
        int i = balance;
        i = i - 1;
        sleep(1);
        balance = i;
        printf("Balance : %d\n",balance);
    }

void* deposit(void* args)
    {
        int i = balance;
        i = i + 1;
        sleep(1);
        balance = i;
        printf("Balance : %d\n",balance);
    }
```

Output 3:

```
Balance : 101
Balance : 99
```