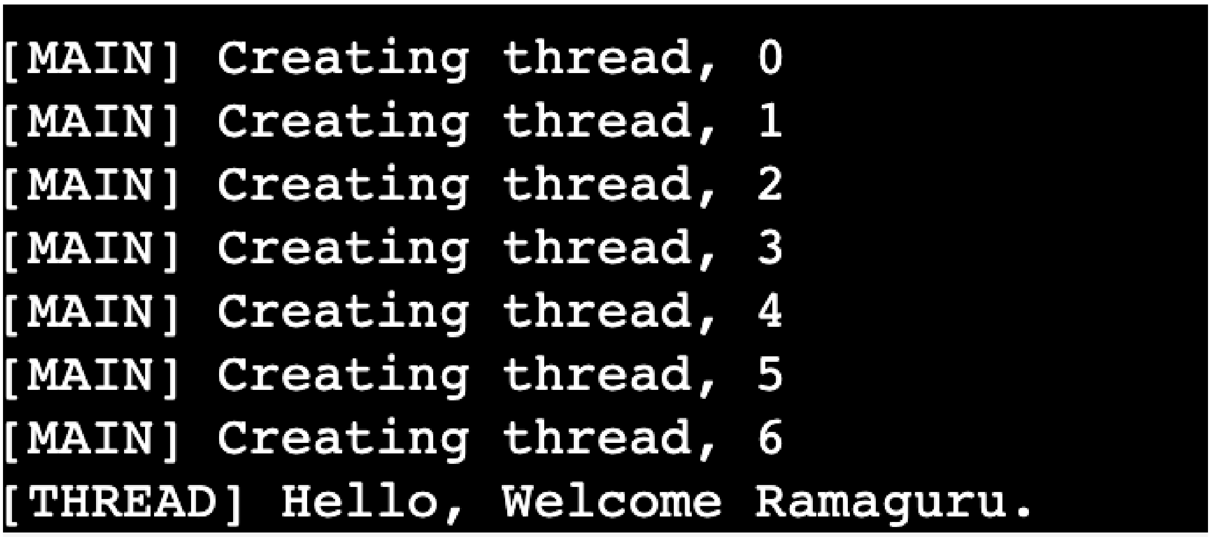
OS LAB
THREADS

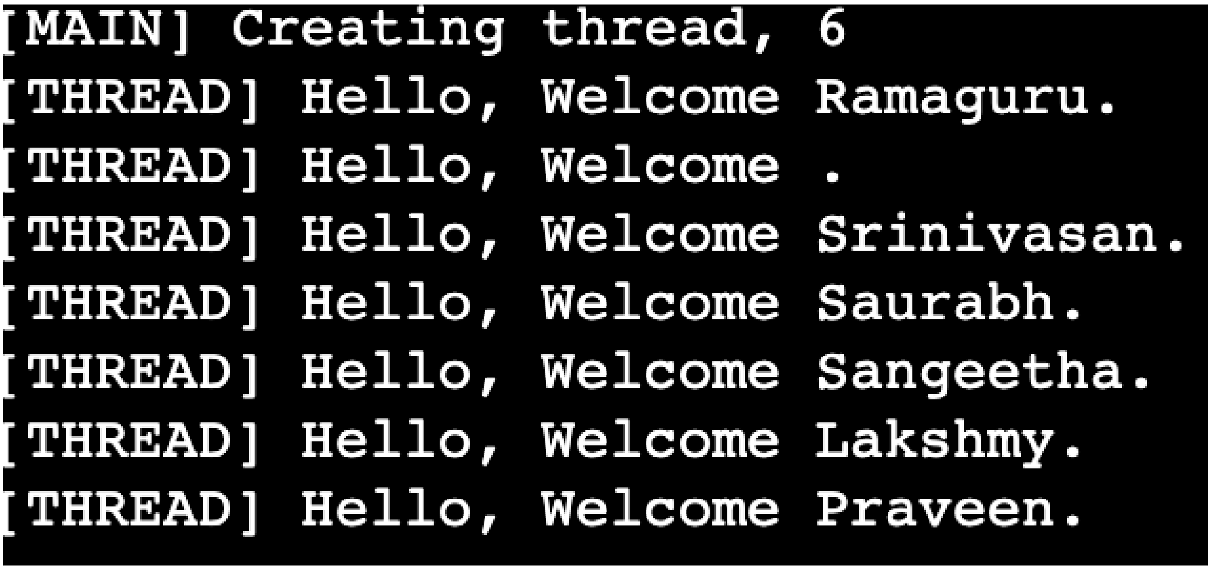
NAME - ANUVARSHINI M K
ROLL NO - CB.EN.U4CYS21008

1)

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

// printWelcomeMessage will be called when the Thread is created in the main function
// which takes string as an argument
void *printWelcomeMessage(void *names) {

   sleep(2);
   char *name = (char *)names;
   printf("\n[THREAD] Hello, Welcome %s.", name);
   pthread_exit(NULL);

}

int main () {

   // thread defintion
   pthread_t threads[5];

   // parameter to be passed to the called function - printWelcomeMessage
   char names[10][15] =
{"Amritha","Praveen","Saurabh","Sangeetha","Lakshmy","Sriniv
asan","Ramaguru"};

   int result;

   for(int i = 0; i < 7; i++ ) {

      printf("\n[MAIN] Creating thread, %d", i);

      // Creating the threading and thus calling the function with
parameter passed to it
      result = pthread_create(&threads[i], NULL,
printWelcomeMessage, (void *)names[i]);

      if (result) {

         printf("Error in creating thread, %d ", result);
         exit(-1);
      }

  }

  // Exit the thread
   pthread_exit(NULL);
}
```

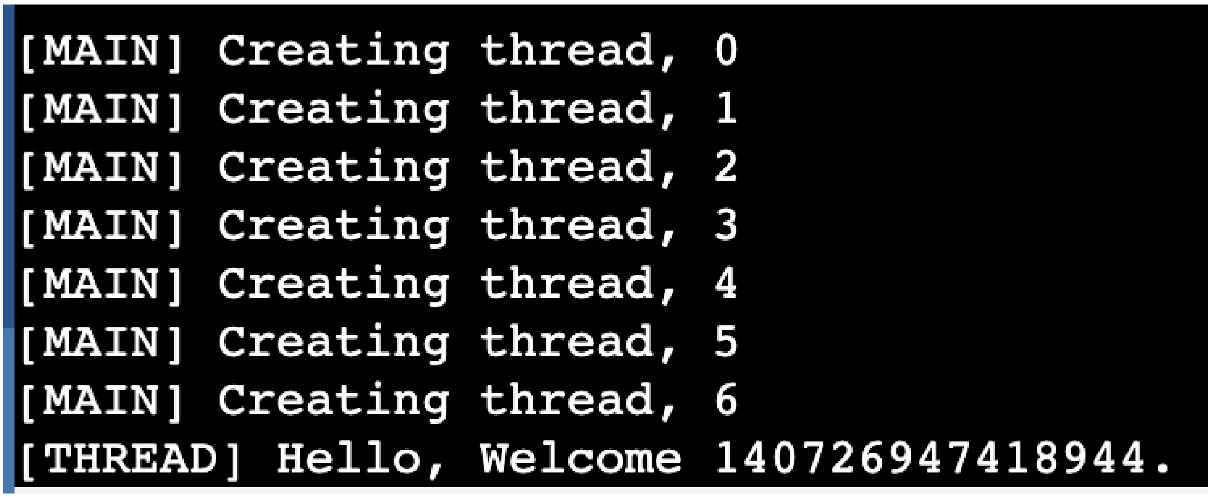


```
[MAIN] Creating thread, 0
[MAIN] Creating thread, 1
[MAIN] Creating thread, 2
[MAIN] Creating thread, 3
[MAIN] Creating thread, 4
[MAIN] Creating thread, 5
[MAIN] Creating thread, 6
[THREAD] Hello, Welcome Ramaguru.
```

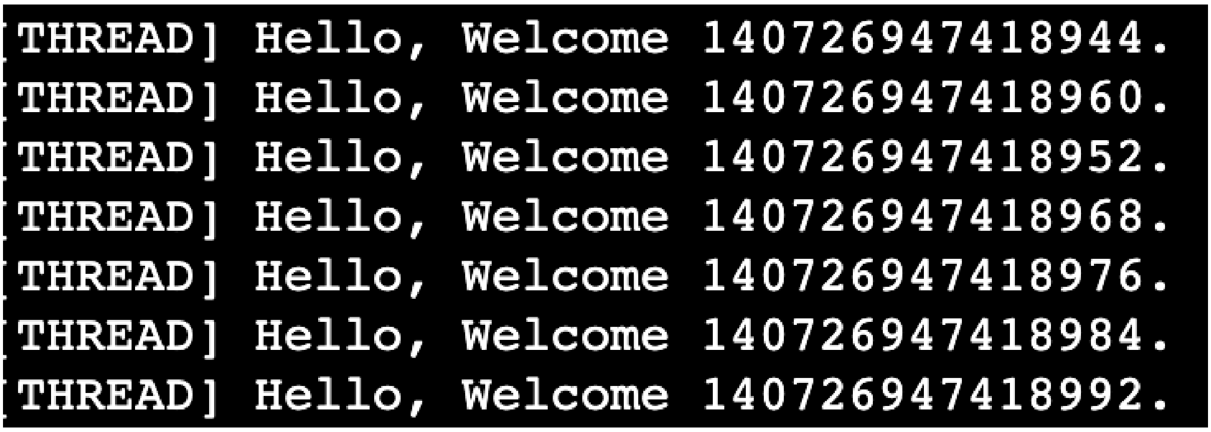


```
[MAIN] Creating thread, 6
[THREAD] Hello, Welcome Ramaguru.
[THREAD] Hello, Welcome .
[THREAD] Hello, Welcome Srinivasan.
[THREAD] Hello, Welcome Saurabh.
[THREAD] Hello, Welcome Sangeetha.
[THREAD] Hello, Welcome Lakshmy.
[THREAD] Hello, Welcome Praveen.
```

```
2) void *printWelcomeMessage(void *threadid) {

   sleep(2);
   long tid = (long)threadid;
   printf("\n[THREAD] Hello, Welcome %ld.", tid);
   pthread_exit(NULL);

}

int main () {

   // thread defintion
   pthread_t threads[5];

   // parameter to be passed to the called function - printWelcomeMessage
   char names[10][15] = {"Amritha","Praveen","Saurabh","Sangeetha","Lakshmy","Srinivasan","Ramaguru"};

   int result;

   for(int i = 0; i < 7; i++ ) {

      printf("\n[MAIN] Creating thread, %d", i);

      // Creating the threading and thus calling the function with parameter passed to it
      result = pthread_create(&threads[i], NULL, printWelcomeMessage,(void *)&threads[i]);

      if (result) {
        printf("Error in creating thread, %d ", result);
        exit(-1);
      }

    }

   // Exit the thread
    pthread_exit(NULL);
}
```



```
[MAIN] Creating thread, 0
[MAIN] Creating thread, 1
[MAIN] Creating thread, 2
[MAIN] Creating thread, 3
[MAIN] Creating thread, 4
[MAIN] Creating thread, 5
[MAIN] Creating thread, 6
[THREAD] Hello, Welcome 140726947418944.
```

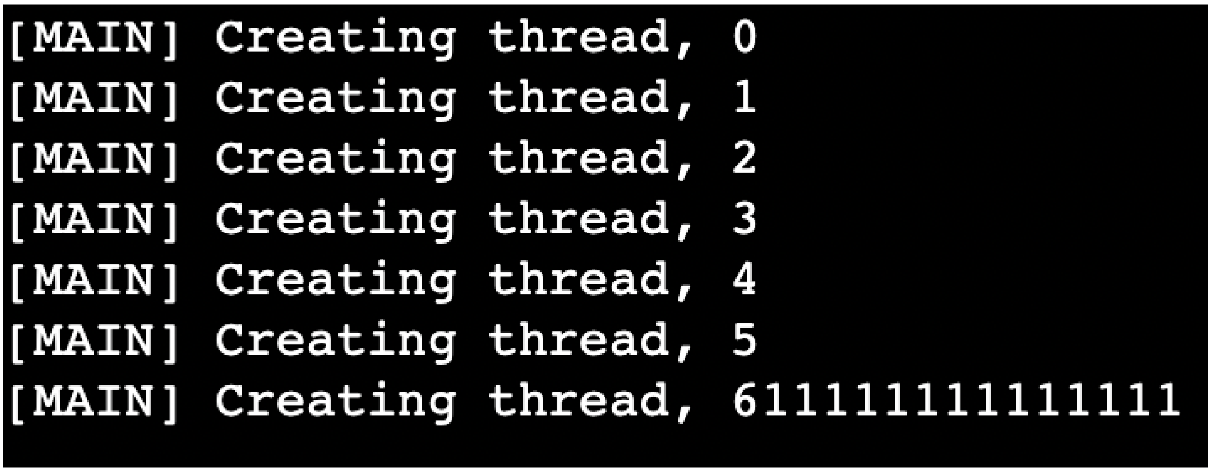


```
[THREAD] Hello, Welcome 140726947418944.
[THREAD] Hello, Welcome 140726947418960.
[THREAD] Hello, Welcome 140726947418952.
[THREAD] Hello, Welcome 140726947418968.
[THREAD] Hello, Welcome 140726947418976.
[THREAD] Hello, Welcome 140726947418984.
[THREAD] Hello, Welcome 140726947418992.
```

3)

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


struct argfunc{
    int a;
    int b;
    char name[10];
};
void *addition(void *arg) {

   struct argfunc *obj = arg;
   int c = obj->a + obj->b;
   printf("%d",c);

}

int main () {

   // thread defintion
   pthread_t threads[5];

   // parameter to be passed to the called function -
printWelcomeMessage


   int result;
   struct argfunc mobj;
   mobj.a = 5;
```

```
  mobj.b = 6;


  for(int i = 0; i < 7; i++ ) {

    printf("\n[MAIN] Creating thread, %d", i);

    // Creating the threading and thus calling the function with parameter passed to it
    result = pthread_create(&threads[i], NULL, addition,&mobj);

    if (result) {

      printf("Error in creating thread, %d ", result);
      exit(-1);
    }

  }

  // Exit the thread
  pthread_exit(NULL);
}
```



```
[MAIN] Creating thread, 0
[MAIN] Creating thread, 1
[MAIN] Creating thread, 2
[MAIN] Creating thread, 3
[MAIN] Creating thread, 4
[MAIN] Creating thread, 5
[MAIN] Creating thread, 611111111111111
```