```
#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","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 *)names[i]);

    if (result) {

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

  }

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

main.c

```c
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
void *printWelcomeMessage(void *names) {
   sleep(5);
   char *name = (char *)names;
   printf("\n[THREAD] Hello, Welcome %s.", name);
   pthread_exit(NULL);

}

int main () {

   // thread defintion
   pthread_t threads[7];

   // 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 *)names[i]);

      if (result) {

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

   }
   pthread_exit(NULL);
}
```

input

```
[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.
[THREAD] Hello, Welcome Srinivasan.
[THREAD] Hello, Welcome Lakshmy.
[THREAD] Hello, Welcome Saurabh.
[THREAD] Hello, Welcome Amritha.
[THREAD] Hello, Welcome Praveen.
[THREAD] Hello, Welcome Sangeetha.

...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include <pthread.h>

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

void *addition() {

   int value1;
```

```
    int value2;
    printf("Enter the 1st element:");
    scanf("%d",&value1);
    printf("Enter the 2nd element:");
    scanf("%d",&value2);
    int total = value1 + value2;
    printf("%d",total);

}

int main () {

   // thread defintion
   pthread_t threads;
   int result;


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

      if (result) {

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


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

}

main.c

```c
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
void *addition() {
    int value1;
    int value2;
    printf("Enter the 1st element:");
    scanf("%d",&value1);
    printf("Enter the 2nd element:");
    scanf("%d",&value2);
    int total = value1 + value2;
    printf("%d",total);

}

int main () {

   // thread defintion
   pthread_t threads;
   int result;


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

     if (result) {

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


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