

```

#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
1 #include <pthread.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 void *printWelcomeMessage(void *names) {
6     sleep(5);
7     char *name = (char *)names;
8     printf("\n[THREAD] Hello, Welcome %s.", name);
9     pthread_exit(NULL);
10 }
11
12
13 int main () {
14     // thread definition
15     pthread_t threads[7];
16
17     // parameter to be passed to the called function - printWelcomeMessage
18     char names[10][15] = {"Amritha", "Praveen", "Saurabh", "Sangeetha", "Lakshmy", "Srinivasan", "Ramaguru"};
19
20     int result;
21
22     for(int i = 0; i < 7; i++) {
23         printf("\n[MAIN] Creating thread, %d", i);
24
25         // Creating the threading and thus calling the function with parameter passed to it
26         result = pthread_create(&threads[i], NULL, printWelcomeMessage, (void *)names[i]);
27
28         if (result) {
29             printf("Error in creating thread, %d ", result);
30             exit(-1);
31         }
32     }
33     pthread_exit(NULL);
34 }

```

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.

```

```
#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
1 #include <pthread.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4 #include <unistd.h>
5 void *addition() {
6     int value1;
7     int value2;
8     printf("Enter the 1st element:");
9     scanf("%d",&value1);
10    printf("Enter the 2nd element:");
11    scanf("%d",&value2);
12    int total = value1 + value2;
13    printf("%d",total);
14 }
15 }
16
17 int main () {
18
19     // thread definition
20     pthread_t threads;
21     int result;
22
23
24     // Creating the threading and thus calling the function with parameter passed to it
25     result = pthread_create(&threads, NULL, addition, NULL);
26
27     if (result) {
28
29         printf("Error in creating thread, %d ", result);
30         exit(-1);
31     }
32
33     // Exit the thread
34     pthread_exit(NULL);
35 }
36 }
```

Enter the 1st element:23  
Enter the 2nd element:22  
45  
  
...Program finished with exit code 0  
Press ENTER to exit console.