

30. Write C programs to demonstrate the following thread related concepts.

(i) create (ii) join (iii) equal (iv) exit

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

void* threadFunc(void* arg) {
    printf("Inside thread function. Thread ID: %lu\n", pthread_self());
    pthread_exit(NULL);
}

int main() {
    pthread_t t1, t2;

    if (pthread_create(&t1, NULL, threadFunc, NULL) != 0) {
        printf("Failed to create thread 1\n");
        return 1;
    }

    if (pthread_create(&t2, NULL, threadFunc, NULL) != 0) {
        printf("Failed to create thread 2\n");
        return 1;
    }

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
}
```

```

if (pthread_equal(t1, t2))
    printf("Threads are equal.\n");
else
    printf("Threads are NOT equal.\n");
printf("Main thread completed.\n");
pthread_exit(NULL);
}

```

OUTPUT-

```

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

void* threadFunc(void* arg) {
    printf("Inside thread function. Thread ID: %lu\n", pthread_self());
    pthread_exit(NULL);
}

int main() {
    pthread_t t1, t2;

    if (pthread_create(&t1, NULL, threadFunc, NULL) != 0) {
        printf("Failed to create thread 1\n");
        return 1;
    }

    if (pthread_create(&t2, NULL, threadFunc, NULL) != 0) {
        printf("Failed to create thread 2\n");
        return 1;
    }

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    if (pthread_equal(t1, t2))
        printf("Threads are equal.\n");
    else
        printf("Threads are NOT equal.\n");
    printf("Main thread completed.\n");
    pthread_exit(NULL);
}

```

Inside thread function. Thread ID: 1
Inside thread function. Thread ID: 2
Threads are NOT equal.
Main thread completed.

Process exited after 2.093 seconds with return value 0
Press any key to continue . . .