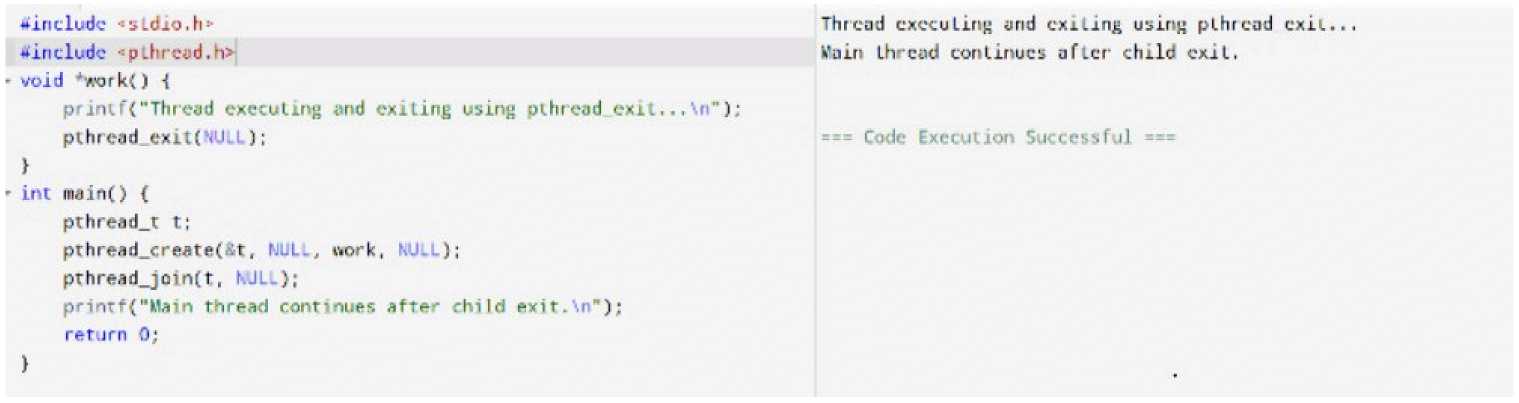
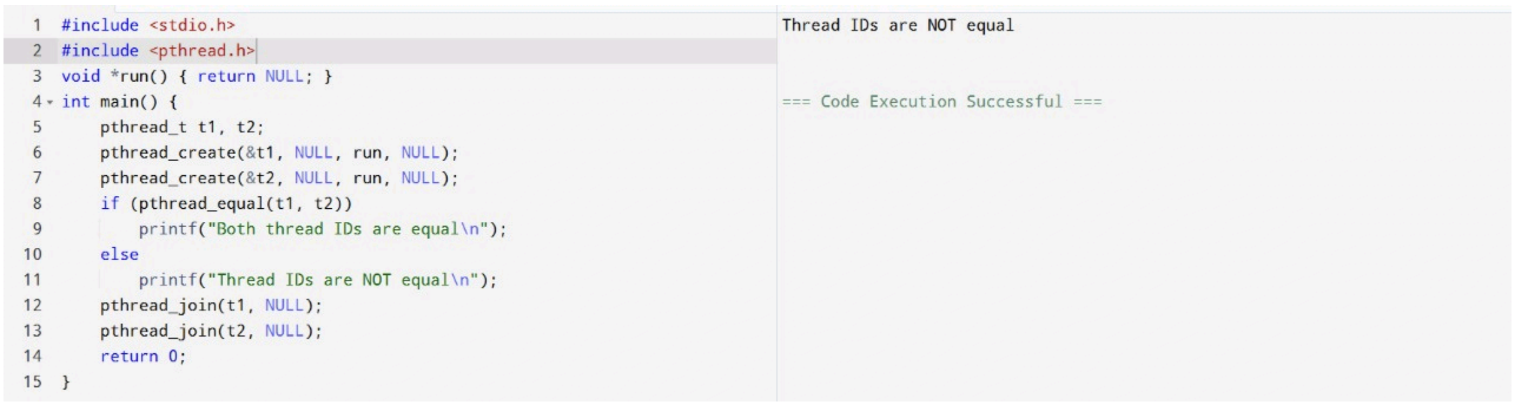


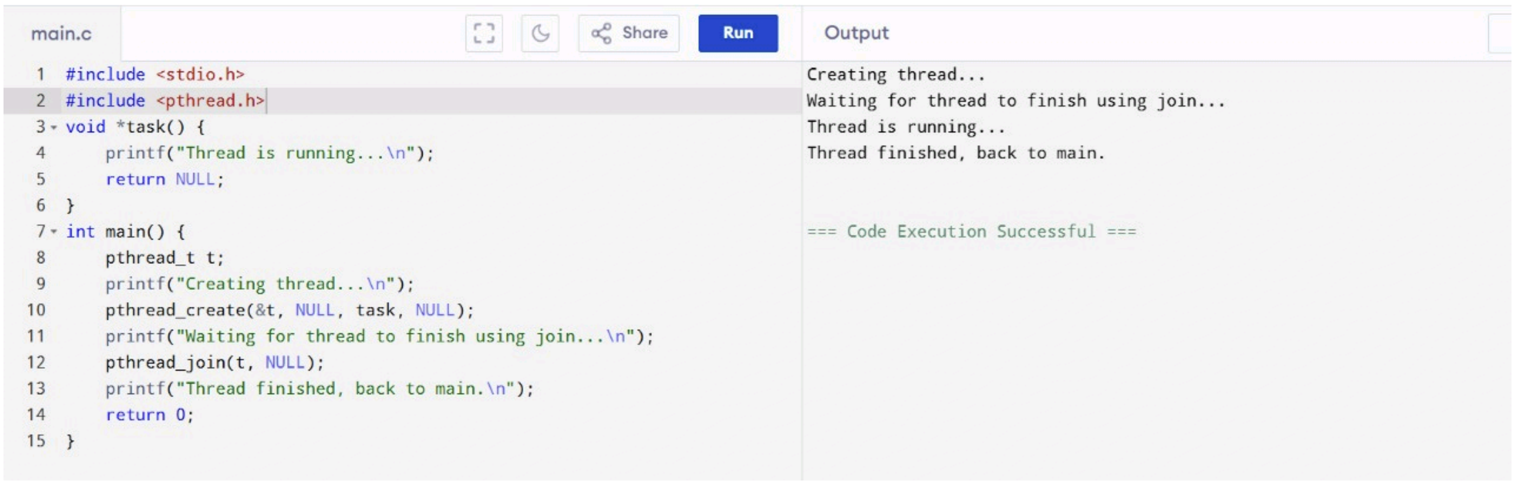
```c
#include <stdio.h>
#include <pthread.h>
void *work() {
    printf("Thread executing and exiting using pthread_exit...\n");
    pthread_exit(NULL);
}
int main() {
    pthread_t t;
    pthread_create(&t, NULL, work, NULL);
    pthread_join(t, NULL);
    printf("Main thread continues after child exit.\n");
    return 0;
}
```

```
Thread executing and exiting using pthread exit...
Main thread continues after child exit.


=== Code Execution Successful ===
```



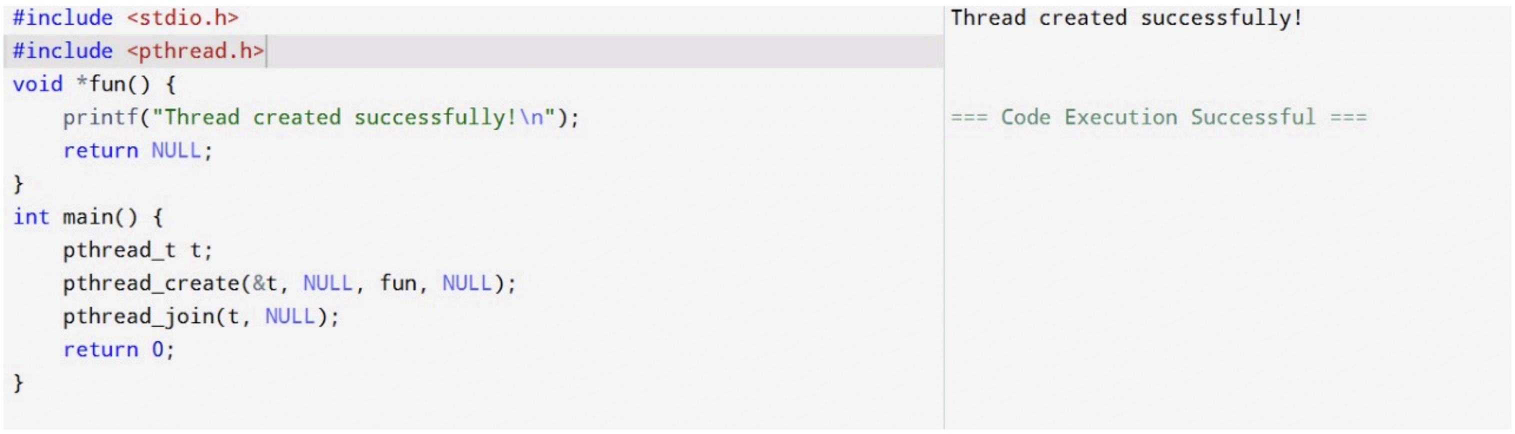
```c
#include <stdio.h>
#include <pthread.h>
void *run() { return NULL; }
int main() {
    pthread_t t1, t2;
    pthread_create(&t1, NULL, run, NULL);
    pthread_create(&t2, NULL, run, NULL);
    if (pthread_equal(t1, t2))
        printf("Both thread IDs are equal\n");
    else
        printf("Thread IDs are NOT equal\n");
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    return 0;
}
```

```
Thread IDs are NOT equal


=== Code Execution Successful ===
```



```c
#include <stdio.h>
#include <pthread.h>
void *task() {
    printf("Thread is running...\n");
    return NULL;
}
int main() {
    pthread_t t;
    printf("Creating thread...\n");
    pthread_create(&t, NULL, task, NULL);
    printf("Waiting for thread to finish using join...\n");
    pthread_join(t, NULL);
    printf("Thread finished, back to main.\n");
    return 0;
}
```

```
Creating thread...
Waiting for thread to finish using join...
Thread is running...
Thread finished, back to main.


=== Code Execution Successful ===
```



```c
#include <stdio.h>
#include <pthread.h>
void *fun() {
    printf("Thread created successfully!\n");
    return NULL;
}
int main() {
    pthread_t t;
    pthread_create(&t, NULL, fun, NULL);
    pthread_join(t, NULL);
    return 0;
}
```

```
Thread created successfully!


=== Code Execution Successful ===
```