11. Illustrate the concept of multithreading using a C program.

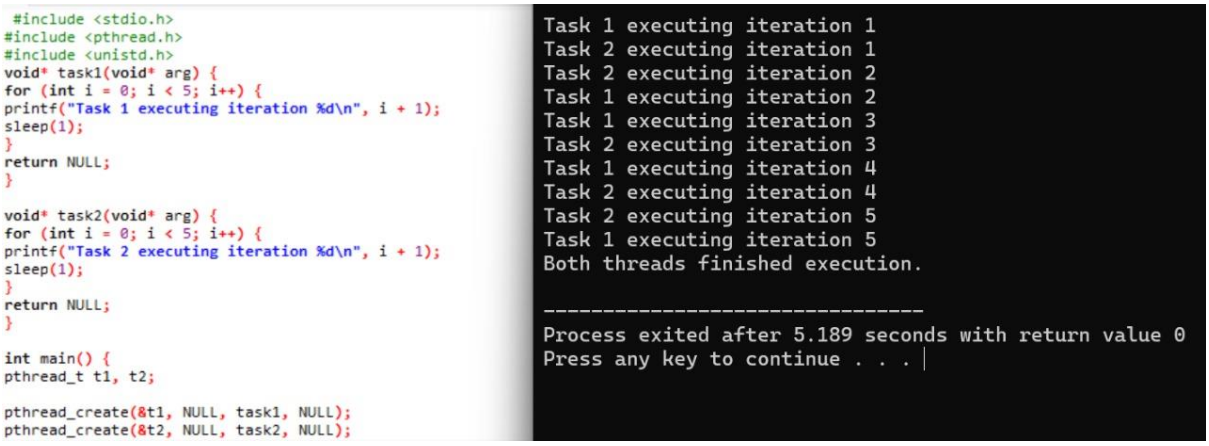
```
 #include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* task1(void* arg) {
for (int i = 0; i < 5; i++) {
printf("Task 1 executing iteration %d\n", i + 1);
sleep(1);
}
return NULL;
}

void* task2(void* arg) {
for (int i = 0; i < 5; i++) {
printf("Task 2 executing iteration %d\n", i + 1);
sleep(1);
}
return NULL;
}

int main() {
pthread_t t1, t2;

pthread_create(&t1, NULL, task1, NULL);
pthread_create(&t2, NULL, task2, NULL);

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

printf("Both threads finished execution.\n");

return 0;

}

OUTPUT:



```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* task1(void* arg) {
for (int i = 0; i < 5; i++) {
printf("Task 1 executing iteration %d\n", i + 1);
sleep(1);
}
return NULL;
}

void* task2(void* arg) {
for (int i = 0; i < 5; i++) {
printf("Task 2 executing iteration %d\n", i + 1);
sleep(1);
}
return NULL;
}

int main() {
pthread_t t1, t2;

pthread_create(&t1, NULL, task1, NULL);
pthread_create(&t2, NULL, task2, NULL);
```

```
Task 1 executing iteration 1
Task 2 executing iteration 1
Task 2 executing iteration 2
Task 1 executing iteration 2
Task 1 executing iteration 3
Task 2 executing iteration 3
Task 1 executing iteration 4
Task 2 executing iteration 4
Task 2 executing iteration 5
Task 1 executing iteration 5
Both threads finished execution.

--------------------------------
Process exited after 5.189 seconds with return value 0
Press any key to continue . . .
```