

1.

Source Code:

```
// COSC 519 Homework 4 Problem 1
// Create a Thread that waits for keyboard input or escape character and then returns
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

void *runner(void *param); //threads call this function

int main()
{
    pthread_t tid; //thread identifier
    pthread_attr_t attr; //thread attributes

    pthread_attr_init(&attr); //get default attributes
    pthread_create(&tid, &attr, runner, 0); //create the thread

    //Wait for thread to exit
    pthread_join(tid, NULL);
    printf("Waited for Thread and it returned.\n");
}

void *runner(void *param)
{
    int c;
    do{
        printf("Type a Character, or press ESC to exit: ");
        c = getchar();
        printf("\n");
    }while((c != 27) && (c != 120)); //Escape Character or x in decimal
    return(0);
}
```

Output:

Type a Character, or press ESC to exit: x

Waited for Thread and it returned.

2.

Source Code:

```
// COSC 519 Homework 4 Problem 2
// Create 10 Threads that print hello and return
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

void *runner(void *param); //threads call this function
int NUM_THREADS = 10;

int main()
{

    int i,j;
    pthread_t tid;

    // Create NUM_THREADS Threads
    for (i = 0; i < NUM_THREADS; i++)
        pthread_create(&tid, NULL, runner, (void *)&tid);

    //Wait for all threads to exit
    for (j=0;j<NUM_THREADS;j++){
        pthread_join(tid, NULL);
    }
    printf("Waited for all threads.\n");
}

void *runner(void *param)
{
    printf("Hello from Thread: %ld\n", pthread_self());
    return(0);
}
```

Output:

```
Hello from Thread: 140472063063808
Hello from Thread: 140472054671104
Hello from Thread: 140472046278400
Hello from Thread: 140472037885696
Hello from Thread: 140472029492992
Hello from Thread: 140472021100288
Hello from Thread: 140471938578176
Hello from Thread: 140471930185472
```

Hello from Thread: 140471921792768

Hello from Thread: 140471913400064

Waited for all threads.