

```

#include <pthread.h>
#include <iostream>

using namespace std;

// Declare and initialize the global array of greetings
const char* my_messages[4] = {"Italian: Ciao!", "English: Hello!", "Hindi: Namaste!", "Spanish: Hola!"};

// Define the pthread function to print messages
void *printMessage(void *arg) {
    // Cast the void pointer to an integer pointer and dereference to get the index
    int index = *(int*)arg;
    // Print the message at the given index
    cout << my_messages[index] << endl;
    return 0;
}

int main()
{
    // Array to hold thread identifiers
    pthread_t threads[4];
    // Array to hold indices for each thread
    int indices[4] = {0, 1, 2, 3};
    int rc;

    // Create threads for each message
    for (int i = 0; i < 4; ++i) {
        rc = pthread_create(&threads[i], NULL, printMessage, (void*)&indices[i]);
        if (rc) {
            cout << "ERROR; return code from pthread_create() is " << rc << endl;
            return -1;
        }
    }

    // Wait for all threads to complete
    for (int i = 0; i < 4; ++i) {
        pthread_join(threads[i], NULL);
    }

    // Exit the main thread
    pthread_exit(0);
}

```

---

```

#include <pthread.h>
#include <iostream>
#include <unistd.h>

```

```

using namespace std;

```

```

// This is the array that contains the integer values,
// that will be used for the functions,
// countNegative, average, and reverse.

```

```

int arr[10];

// This function is used to count negatives within the data array.
// The function has a return type of void and the parameter is a void pointer.
void *countNegatives(void *arg)
{
    int total = 0;

    for (int index = 0; index < 10; index++)
    {
        if (arr[index] < 0)
            total++;
    }

    cout << "Total negative numbers: "<< total << endl << endl;

    pthread_exit(0);
}

// This function is used to calculate the average value of the data array.
// The function has a return type of void and the parameter is a void pointer.
void *average(void *arg)
{
    double total = 0.0;

    for (int index = 0; index < 10; index++)
    {
        total += arr[index];
    }

    cout << "Average: "<< total / 10.0 << endl << endl;

    pthread_exit(0);
}

// This function is used to print the data array in reverse.
// The function has a return type of void and the parameter is a void pointer.
void *reverse(void *arg)
{
    cout << "The numbers in reverse: " << endl;

    for (int index = 9; index >= 0; index--)
    {
        cout << arr[index] << endl ;
    }

    pthread_exit(0);
}

int main()
{
    // TODO:

```

```

// Add necessary variable declarations.

// TODO:
// Add code to perform any needed initialization
// or to process user input
cout << "Enter 10 integers: ";
for (int i = 0; i < 10; ++i) {
    cin >> arr[i];
}

// TODO: Modify according to assignment requirements
// Create thread(s) that will execute the functions of countNegative, average, and reverse
// and check for the return values for errors.
pthread_t thread1, thread2, thread3;

pthread_create(&thread1, NULL, countNegatives, NULL);
usleep(1000);
pthread_create(&thread2, NULL, average, NULL);
usleep(1000);
pthread_create(&thread3, NULL, reverse, NULL);

pthread_join(thread1, NULL);
pthread_join(thread2, NULL);
pthread_join(thread3, NULL);

return 0;

// NOTE: Using exit here will immediately end execution of all threads
pthread_exit(0);
}

```

---

```

#include <pthread.h>
#include <iostream>

using namespace std;

void *PrintHello(void *arg)
{
    int actual_arg = *((int*) arg);
    std::cout << "Hello World from thread with arg: " << actual_arg << "!\n";
    return 0;
}

int main()
{
    pthread_t id;
    int rc;
    std::cout << "In main: creating thread \n";

    int t;
    cout << "Enter an integer: ";

```

```
cin >> t;
rc = pthread_create(&id, NULL, PrintHello, (void*) &t);

if (rc){
    std::cout << "ERROR; return code from pthread_create() is " << rc << std::endl;
    return -1;
}

pthread_exit(0);
}
```