EXP 13: Illustrate the concept of multithreading using a C program.

```
#include <stdio.h>

// Memory blocks and processes
int blocks[10], block_status[10] = {0}; // 0 = free, 1 = allocated
int processes[10], process_allocation[10] = {0}; // which block each process is allocated to
int m, p; // Number of blocks and processes

void display() {
   printf("\nMemory Blocks:\n");
   printf("Block\tSize\tStatus\n");
   int i; // Declare loop variable at start for older C standards
   for(i = 0; i < m; i++) {
      printf("%d\t%d\t", i+1, blocks[i]);
      if(block_status[i] == 0) {
         printf("Free\n");
      } else {
         printf("Allocated to P%d\n", block_status[i]);
      }
   }
}

void reset() {
   int i; // Declare loop variable at start
   for(i = 0; i < m; i++) block_status[i] = 0;
   for(i = 0; i < p; i++) process_allocation[i] = 0;
}

void first_fit() {
   reset();
```

```
    int i, j; // Declare loop variables at start
    for(i = 0; i < p; i++) {
        for(j = 0; j < m; j++) {
            if(block_status[j] == 0 && blocks[j] >= processes[i]) {
                block_status[j] = i+1; // Mark as allocated to process i+1
                process_allocation[i] = j+1; // Process allocated to block j+1
                break;
            }
        }
    }
    printf("\nFirst Fit Allocation:");
    display();
}

int main() {
    int i; // Declare loop variable at start

    printf("Enter number of memory blocks (max 10): ");
    scanf("%d", &m);
    printf("Enter size of each block:\n");
    for(i = 0; i < m; i++) {
        printf("Block %d: ", i+1);
        scanf("%d", &blocks[i]);
    }

    printf("Enter number of processes (max 10): ");
    scanf("%d", &p);
    printf("Enter size of each process:\n");
    for(i = 0; i < p; i++) {
        printf("Process %d: ", i+1);
        scanf("%d", &processes[i]);
```

```
    }

    first_fit();

    return 0;
}
```

Sample Input

Enter number of memory blocks (max 10): 5

Enter size of each block:

Block 1: 55

Block 2: 33

Block 3: 54

Block 4: 3

Block 5: 5

Enter number of processes (max 10): 6

Enter size of each process:

Process 1: 33

Process 2: 4

Process 3: 55

Process 4: 34

Process 5: 34

Process 6: 45

Sample Output