```
// From James Peckol, Embedded Systems
// A simple OS kernel - step 1
#include <stdio.h>

// Prototypes for the tasks
void get (void* aNumber);               // input task
void increment (void* aNumber);         // computation task
void display (void* aNumber);    // display task

void main(void) {
  int i=0;                              // queue index
  int data;                             // declare a shared var, data
  int* aPtr = &data;                    // point to it

  void(*queue[3])(void*);               // declare queue as an array of pointers to
                                        // fns taking an arg of type void*
  queue[0]=get;                         // enter the tasks into the queue
  queue[1]=increment;
  queue[2]=display;

  while(1) {
    queue[i]((void*)aPtr);              // dispatch each task in turn
    i=(i+1)%3;
  }
  return;
}

void get(void* aNumber) {         // perform input operation
  printf("Enter a number, 0..9 ");
  *(int*) aNumber = getchar();
  getchar();                            // discard CR
  *(int*) aNumber -= '0';               // convert to decimal from ASCII
  return;
}

void increment(void* aNumber) {// perform computation
  int* aPtr = (int*) aNumber;
  (*aPtr)++;
  return;
}

void display (void* aNumber) {   // perform output operation
  printf("The result is: %d\n",*(int*)aNumber);
  return;
}
```