

```

    for (;;) {
        printf("Enter operation code: ");
        scanf(" %c", &code);
        while (getchar() != '\n') /* skips to end of line */
            ;
        switch (code) {
            case 'i': insert();
                        break;
            case 's': search();
                        break;
            case 'u': update();
                        break;
            case 'p': print();
                        break;
            case 'q': return 0;
            default: printf("Illegal code\n");
        }
        printf("\n");
    }
}

/*****
 * find_part: Looks up a part number in the inventory
 *            array. Returns the array index if the part
 *            number is found; otherwise, returns -1.
 *****/
int find_part(int number)
{
    int i;

    for (i = 0; i < num_parts; i++)
        if (inventory[i].number == number)
            return i;
    return -1;
}

/*****
 * insert: Prompts the user for information about a new
 *         part and then inserts the part into the
 *         database. Prints an error message and returns
 *         prematurely if the part already exists or the
 *         database is full.
 *****/
void insert(void)
{
    int part_number;

    if (num_parts == MAX_PARTS) {
        printf("Database is full; can't add more parts.\n");
        return;
    }

    printf("Enter part number: ");
    scanf("%d", &part_number);

```