```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define N 1000

/* Prototypes */
void borrowBook(int);
void viewBooks(int);
void returnBook(int);

/* Function pointer */
void (*fptr[])(int) = {borrowBook, viewBooks, returnBook};

/* Student struct */
struct student
{
    int stdNum;
    int isbnNum;
    char bookTitle[N];
}s[N];


int main(void)
{
    int num;
    char defaultPsswrd[] = "Admin123";
    int loginAttempt = 0, option;
    char password[N], answr[N];

    /* Clear ouput screen */
    system("clear");

    jump:
    while (loginAttempt < 3)
    {
        printf("Password: ");
        scanf("%s", password);

        if (strcmp(password, defaultPsswrd) == 0)
        {
            printf("\n*****************************\n");
            printf("DKUT Library Management System\n");
            printf("*****************************\n");
            printf("1. Borrow a Book\n");
            printf("2. View borrowed books\n");
            printf("3. Return a Book\n");
            printf("4. Quit\n");
            break;
        }
        else
            loginAttempt++;
```

```c
    }

    if (loginAttempt == 3)
    {
        system("clear"); // Clear screen
        printf("You have exhausted your number of login");
    }
    else
    {
        // Choose from 1-4
        scanf("%d", &option);
        system("clear"); // Clear screen

        /* TODO: Bound checking if the option > 4 */

        if ((option-1) == 0)
        {
            printf("How many student?\n->: ");
            scanf("%d", &num);
            fptr[0](num);
            printf("Do you want to issue another book?: ");
            scanf("%s", answr);
            system("clear"); // clear screen
            if (strcmp(answr, "y") == 0 | strcmp(answr, "Y") == 0)
                goto jump;
        }
        else if (option == 4)
            exit;
        else
            /* Function pointer */
            fptr[option-1](num);
    }
    return (0);
}

/*
 * borrowBook - Store arrays of Student Number, ISBN, Book Title.
 * @num: Number of items to be stored.
 */
void borrowBook(int num)
{
    printf("Student Number\n");
    printf("---------------\n");
    for (int i = 0; i < num; i++)
    {
        printf("stduent %d: ", i+1);
        scanf("%d", &s[i].stdNum);
    }

    printf("ISBN\n");
    printf("---------------\n");
```

```c
    for (int i = 0; i < num; i++)
    {
        printf("ISBN %d: ", i+1);
        scanf("%d", &s[i].isbnNum);
    }

    printf("Book Title\n");
    printf("----------------\n");
    for (int i = 0; i < num; i++)
    {
        printf("Book Title %d: ", i+1);
        // read & ignore extra characters
        getchar();
        // read string with spaces
        scanf("%[^\n]s", s[i].bookTitle);
    }
}

/*
 * viewBooks - View student record of borrowed books.
 * @num: Number of item.
 */
void viewBooks(int num)
{
    for (int i = 0; i < num; i++)
    {
        printf("\nStudent %d\n", i+1);
        printf("================\n");
        printf("-  Student Number: %d\n", s[i].stdNum);
        printf("-  ISBN: %d\n", s[i].isbnNum);
        printf("-  Book Title: %s\n", s[i].bookTitle);
    }
}

/*
 * returnBook - Checks if a borrowed book is returned or not.
 * @num: Number of item.
 */
void returnBook(int num)
{
    int pISBN, flag, idx;

    printf("ISBN: ");
    scanf("%d", &pISBN);

    /* Bound checking for pISBN */
    /* TODO: Allow Admin to input a valid ISBN */
    if (pISBN > num)
    {
        printf("Not record of ISBN");
        /* TODO: show avaliable records */
```

```
154     }
155     else
156     {
157         for (int i = 0; i < num; i++)
158         {
159             if (pISBN == s[i].isbnNum)
160             {
161                 flag = 1; // 1 = True
162                 idx = i; // Index of the returened item
163                 break;
164             }
165             else
166                 flag = 0; // 0 = False
167         }
168         if (flag == 1)
169         {
170             printf("Return Successfully\n");
171             /* Remove student from list if returened */
172             for (size_t i = idx; i < num-1; i++)
173             {
174                 s[idx] = s[num-1];
175             }
176             num--;
177             printf("Avaliable students\n");
178             printf("------------------------");
179             viewBooks(num);
180         }
181         else
182         {
183             printf("Not returned\n");
184             printf("Avaliable students\n");
185             printf("------------------------");
186             viewBooks(num);
187         }
188     }
189 }
```