

**REPORT FOR COURIER MANAGEMENT SYSTEM PROGRAM USING C LANGUAGE**

AS A PROJECT WORK FOR THE COURSE

**COMPUTER PROGRAMMING (CSE 101)**

**DATE OF SUBMISSION:** 21/04/2023 **SUBMITTED TO:** Dr. Gulshan Kumar

**SECTION:** K22EG **SEMESTER:** Second Semester

|  |  |  |
| --- | --- | --- |
| **Roll No.** | **Name** | **Reg. No.** |
| 21 | Ankit | 12205604 |
| 22 | Mankina Nishanth Sai | 12206439 |
| 23 | Praveen Kumar | 12206512 |
| 24 | Deepika Kumari | 12206537 |

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**LOVELY PROFESSIONAL UNIVERSITY, JALANDHAR,**

**PUNJAB, INDIA**

**CONTRIBUTION OF EACH MEMBER**

**Member 1**

|  |  |
| --- | --- |
| **Name:** | **Ankit Maurya** |
| **Reg. No.:** | **12205604** |
| **Roll No.:** | **21** |
| **Section:** | **K22EG** |
| **Role of the member in current work:** | **Contributed in developing the add courier and display courier module.**  **Also made the project report.** |
|  |  |

**Member 2**

|  |  |
| --- | --- |
| **Name:** | MankinaNishanth Sai |
| **Reg. No.:** | 12206439 |
| **Roll No.:** | **22** |
| **Section:** | **K22EG** |
| **Role of the member in current work:** | **Contributed in developing the update courier and show delivered courier module.** |
|  |  |

**Member 3**

|  |  |
| --- | --- |
| **Name:** | Praveen Kumar |
| **Reg. No.:** | 12206512 |
| **Roll No.:** | **23** |
| **Section:** | **K22EG** |
| **Role of the member in current work:** | **Contributed in developing the sorting the courier module. Also contributed for the update module.** |
|  |  |

**Member 4**

|  |  |
| --- | --- |
| **Name:** | Deepika Kumari |
| **Reg. No.:** | 12206537 |
| **Roll No.:** | **24** |
| **Section:** | **K22EG** |
| **Role of the member in current work:** | **Contributed in developing the login to system, changing pin of the system and track courier module.** |
|  |  |

**DECLARATION**

We declare that the project work reported entitled " COURIER MANAGEMENT SYSTEM ” in partial fulfilment of the requirement for the award of Degree for Bachelors of Technology in CSE at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of our supervisor Ms. Manbir Kaur. The content of this project represents authentic and honest effort conducted, in its entirety, by us. We are fully responsible for the contents of our project work.

**Student Name** **Registration Number**  **Roll No.**

1 Ankit Maurya 12205604 21

2MankinaNishanth Sai12206439 22

3 Praveen Kumar 12206512 23

4 Deepika Kumari 12206537 24

**INDEX**

1. **C introduction 5**
2. **Project Overview 6-7**
3. **Code 8-16**
4. **Results 17-23**
5. **C**

**INTRODUCTION:**

C (pronounced /ˈsiː/ – like the letter c) is a general-purpose computer programming language. It was created in the 1970s by Dennis Ritchie, and remains very widely used and influential. By design, C's features cleanly reflect the capabilities of the targeted CPUs. It has found lasting use in operating systems, device drivers, protocol stacks, though decreasingly for application software. C is commonly used on computer architectures that range from the largest supercomputers to the smallest microcontrollers and embedded systems.

A successor to the programming language B, C was originally developed at Bell Labs by Ritchie between 1972 and 1973 to construct utilities running on Unix. It was applied to re-implementing the kernel of the Unix operating system.[8] During the 1980s, C gradually gained popularity. It has become one of the most widely used programming languages, with C compilers available for practically all modern computer architectures and operating systems. C has been standardized by ANSI since 1989 (ANSI C) and by the International Organization for Standardization (ISO).

C is an imperative procedural language, supporting structured programming, lexical variable scope and recursion, with a static type system. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code.

1. **PROJECT**

**Courier Management System**

There has been a rapid increase in support and attention given to courier management in recent years in the public sector all over the world as people embrace information and communication technologies in their management of their corporate records due to accountability and auditing requirements. There has been a lot of complaints raised by the people especially in the country coming from more developed countries in relation to the manual system used in conventional non-technological courier management systems an example such parcel/envelope misplacement, parcels going missing and this leads to mistrust in the use of this systems and opting for the unorthodox methods. This causes the companies losses due to replacement of the parcels in money form. Courier management computerization is “the incorporation of appropriate technology to help managers manage information. Technology is considered suitable when it utilizes the most abundant domestic possessions and conserves investment and skilled personnel”. The main aim of this project is to computerize the maintenance of courier management using C language as our programming language. We have divided our project objectives into several modules created by several team members in order to increase the overall efficiency of the project.

Modules:

1. Adding courier
2. Displaying courier
3. Updating delivery status
4. Tracking courier
5. Sorting(id) courier
6. Show delivered courier
7. For Changing system pin

Module Description:

1. **Add Courier Module**: An insert module in a courier management system is made to let users add new shipments to the system. The insert module often offers user interface that enables the user to enter information about the package, including the sender and recipient's names and shipping details.

2. **Display Courier Module:** A display module lets users view and look for pending packages in a courier management system. A list of the packages that are currently in the system, along with their tracking numbers, and all other shipping details.

3. **Updating Delivery Status Module:** This module allows user to update the delivery status of the courier from pending delivery to delivery done with the help of courier id.

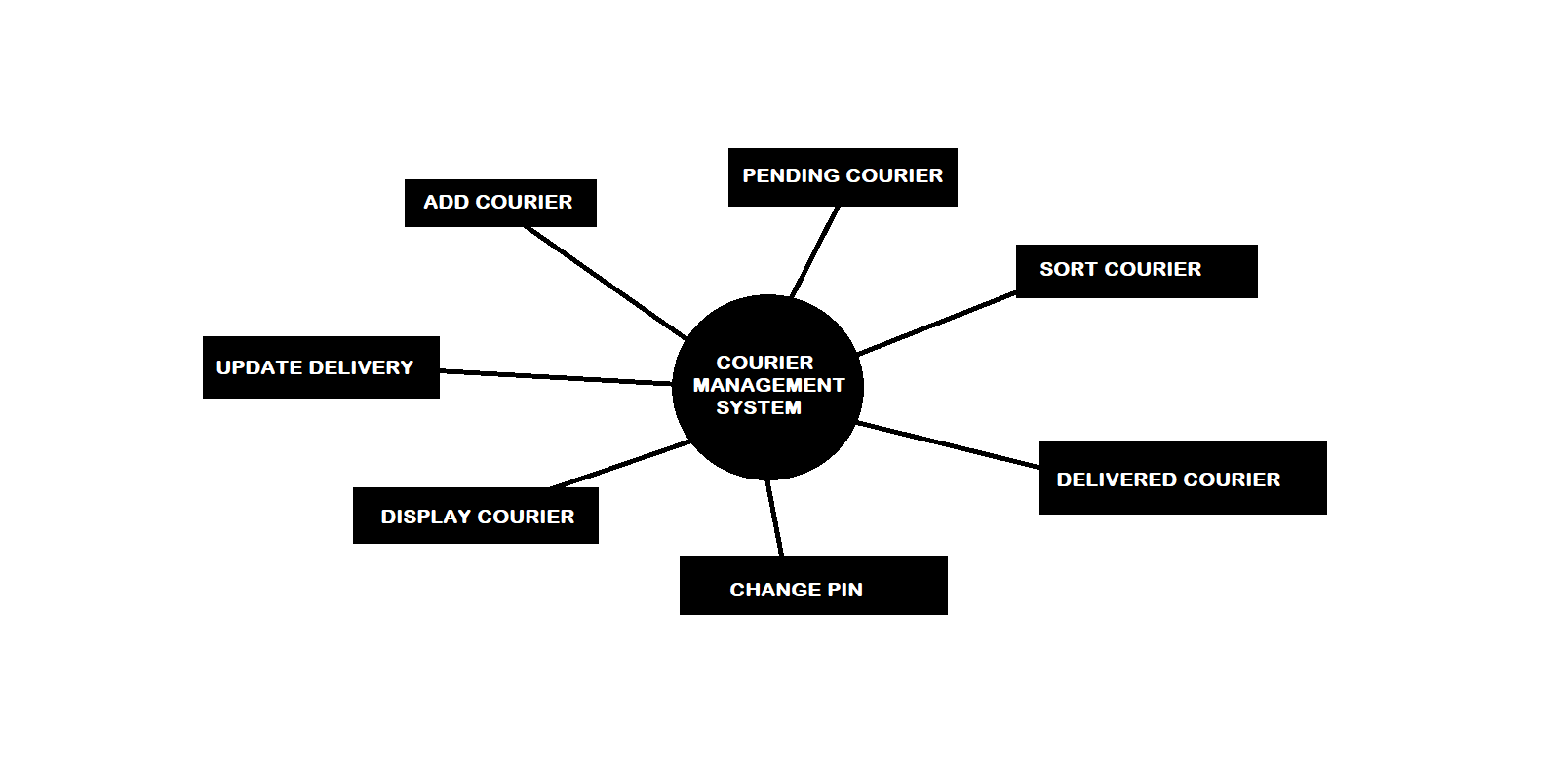
4. **Tracking Courier Module:** A search module of a courier management system that enables users to look for particular items in the system using courier id.

5. **Sorting Courier Module:** This module sort list the all courier whose delivery are pending based on their increasing courier id. It helps user to find any particular courier easily.

6. **Show Delivered Courier Module:** This module displays all the courier that are delivered.

7. **Changing System Pin Module:** This module allows the user to change the current pin of the system.

**DATA FLOW DIAGRAM:**

****

1. **CODE**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct password{ // data-type for storing pin

int pin;

};

struct addcourier{ // data-type for adding courier

int cid;

char cname[30]; // courier name

char sname[30]; // sender name

char rname[30]; // reciever name

int scontact; // sender contact

int rcontact; // reciever contact

char raddress[50]; // reciever address

char odate[20]; // courier order date

char dels[20]; // delivery status

int dcharge; // delivery charge

};

void addcourier(){ // function for adding courier

struct addcourier addc[1];

FILE \*fp;

fp=fopen("courierdata.txt","a"); // file for storing courier data

printf(">> Enter courier id: ");

scanf("%d",&addc[0].cid);

printf(">> Enter courier name: ");

scanf("%s",addc[0].cname);

printf(">> Enter sender name(first): ");

scanf("%s",addc[0].sname);

printf(">> Enter reciever name(first): ");

scanf("%s",addc[0].rname);

printf(">> Enter sender contact no.: ");

scanf("%d",&addc[0].scontact);

printf(">> Enter reciever contact no.: ");

scanf("%d",&addc[0].rcontact);

printf(">> Enter delivery address(area\_city): ");

scanf("%s",addc[0].raddress);

printf(">> Enter courier date(dd/mm/yyyy): ");

scanf("%s",addc[0].odate);

printf(">> Enter delivery status(pending/done): ");

scanf("%s",addc[0].dels);

printf(">> Enter delivery charge(Rs.): ");

scanf("%d",&addc[0].dcharge);

fwrite(&addc[0],sizeof(addc[0]),1,fp);

printf("\n");

printf("\n");

printf(" Courier added successfully\n");

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

printf("\n");

fclose(fp);

}

void showcourier(){ // function for displaying pending courier

struct addcourier addc[1];

FILE \*fp;

fp=fopen("courierdata.txt","r");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" ....Pending Courier....\n");

printf("\n");

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

int value; // for string comparison

value=strcmp(addc[0].dels,"pending");

if(value==0){

printf("-Courier id: %d\n",addc[0].cid);

printf("-Courier name: %s\n",addc[0].cname);

printf("-Sender name(first): %s\n",addc[0].sname);

printf("-Reciever name(first): %s\n",addc[0].rname);

printf("-Sender contact no.: %d\n",addc[0].scontact);

printf("-Reciever contact no.: %d\n",addc[0].rcontact);

printf("-Delivery address: %s\n",addc[0].raddress);

printf("-Courier date(dd/mm/yyyy): %s\n",addc[0].odate);

printf("-Delivery status(pending/done): %s\n",addc[0].dels);

printf("-Delivery charge(Rs.): %d\n",addc[0].dcharge);

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

printf("\n");

}

}

fclose(fp);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void searchcourier(){ // function for tracking courier

int findid; // courier id to track

printf("Enter the courier id: ");

scanf("%d",&findid);

struct addcourier addc[1];

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" ....Search Output....\n");

printf("\n");

FILE \*fp;

fp=fopen("courierdata.txt","r");

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

if(addc[0].cid==findid){

printf("-Courier id: %d\n",addc[0].cid);

printf("-Courier name: %s\n",addc[0].cname);

printf("-Sender name(first): %s\n",addc[0].sname);

printf("-Reciever name(first): %s\n",addc[0].rname);

printf("-Sender contact no.: %d\n",addc[0].scontact);

printf("-Reciever contact no.: %d\n",addc[0].rcontact);

printf("-Delivery address: %s\n",addc[0].raddress);

printf("-Courier date(dd/mm/yyyy): %s\n",addc[0].odate);

printf("-Delivery status(pending/done): %s\n",addc[0].dels);

printf("-Delivery charge(Rs.): %d\n",addc[0].dcharge);

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

printf("\n");

}

}

fclose(fp);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void delcourier(){ // function for displaying delivered courier

struct addcourier addc[1];

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" ....Delivered Courier....\n");

printf("\n");

FILE \*fp;

fp=fopen("courierdata.txt","r");

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

int value;

value=strcmp(addc[0].dels,"done");

if(value==0){

printf("-Courier id: %d\n",addc[0].cid);

printf("-Courier name: %s\n",addc[0].cname);

printf("-Sender name(first): %s\n",addc[0].sname);

printf("-Reciever name(first): %s\n",addc[0].rname);

printf("-Sender contact no.: %d\n",addc[0].scontact);

printf("-Reciever contact no.: %d\n",addc[0].rcontact);

printf("-Delivery address: %s\n",addc[0].raddress);

printf("-Courier date(dd/mm/yyyy): %s\n",addc[0].odate);

printf("-Delivery status(pending/done): %s\n",addc[0].dels);

printf("-Delivery charge(Rs.): %d\n",addc[0].dcharge);

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

printf("\n");

}

}

fclose(fp);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void updatecourier(){ // function for updating delivery status

struct addcourier addc[1];

FILE \*fp;

FILE \*fp2;

fp=fopen("courierdata.txt","r");

fp2=fopen("courierdata2.txt","w"); // file for storing coueier data temporarly

int uid; // courier id to update

printf("Enter courier id to update: ");

scanf("%d",&uid);

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

if(addc[0].cid==uid){

printf("Enter updated delivery status(done): ");

scanf("%s",addc[0].dels);

fwrite(&addc[0],sizeof(addc[0]),1,fp2);

}

else{

fwrite(&addc[0],sizeof(addc[0]),1,fp2);

}

}

fclose(fp);

fclose(fp2);

fp=fopen("courierdata2.txt","r");

fp2=fopen("courierdata.txt","w");

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

fwrite(&addc[0],sizeof(addc[0]),1,fp2);

}

fclose(fp);

fclose(fp2);

printf("\n");

printf("\n");

printf(" Delivery Updated Successfully\n");

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

printf("\n");

}

void sortcourier(){ // function for sorting pending courier based on id

struct addcourier addc[1],temp;

FILE \*fp;

fp=fopen("courierdata.txt","r");

int count=0;

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

int value;

value=strcmp(addc[0].dels,"pending");

if(value==0){

count++;

}

}

fclose(fp);

int sortarr[count]; // array for storing courier id

fp=fopen("courierdata.txt","r");

int index=0;

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

int value;

value=strcmp(addc[0].dels,"pending");

if(value==0){

sortarr[index]=addc[0].cid;

index++;

}

}

fclose(fp);

int i,j;

for(i=0;i<count;i++){ // sorting courier data based on courier id

for(j=0;j<(count-i-1);j++){

if(sortarr[j]>sortarr[j+1]){

int temp;

temp=sortarr[j];

sortarr[j]=sortarr[j+1];

sortarr[j+1]=temp;

}

}

}

FILE \*fptr;

fptr=fopen("courierdatasort.txt","w"); // file for storing sorted courier data

for(i=0;i<count;i++){

fp=fopen("courierdata.txt","r");

while(fread(&addc[0],sizeof(addc[0]),1,fp)){

if(addc[0].cid==sortarr[i]){

fwrite(&addc[0],sizeof(addc[0]),1,fptr);

}

}

fclose(fp);

}

fclose(fptr);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" ....Sorted Pending Courier....\n");

printf("\n");

fptr=fopen("courierdatasort.txt","r");

while(fread(&addc[0],sizeof(addc[0]),1,fptr)){

printf("-Courier id: %d\n",addc[0].cid);

printf("-Courier name: %s\n",addc[0].cname);

printf("-Sender name(first): %s\n",addc[0].sname);

printf("-Reciever name(first): %s\n",addc[0].rname);

printf("-Sender contact no.: %d\n",addc[0].scontact);

printf("-Reciever contact no.: %d\n",addc[0].rcontact);

printf("-Delivery address: %s\n",addc[0].raddress);

printf("-Courier date(dd/mm/yyyy): %s\n",addc[0].odate);

printf("-Delivery status(pending/done): %s\n",addc[0].dels);

printf("-Delivery charge(Rs.): %d\n",addc[0].dcharge);

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

printf("\n");

}

fclose(fptr);

}

int enterpin(){ // function for getting pin stored in file

int access;

struct password p[1];

FILE \*fp;

fp=fopen("cpasswd.txt","r"); // file for storing the pin

fread(&p[0],sizeof(p[0]),1,fp);

fclose(fp);

access=p[0].pin;

return access;

}

void changepin(){ // function for storing the updated pin

int task=0;

struct password p[1];

int upass; // upass: current pin as input

FILE \*fp;

fp=fopen("cpasswd.txt","r");

printf("Enter the current pin: ");

scanf("%d",&upass); // upass: input for current pin

fread(&p[0],sizeof(p[0]),1,fp);

if(p[0].pin==upass){

task=1;

}

else{

printf("Invalid Pin, Try Again....\n");

}

fclose(fp);

if(task==1){

fp=fopen("cpasswd.txt","w");

printf("Enter the new pin: ");

scanf("%d",&p[0].pin);

fwrite(&p[0],sizeof(p),1,fp);

fclose(fp);

printf("\n");

printf(" Pin changed successfully\n");

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

printf("\n");

}

}

int main(){

printf("################################################\n");

printf("------------- LPU COURIER SERVICES -------------\n");

printf("\n");

int epass,apass; // epass pin input from user, apass system pin

apass=enterpin(); // calling function to get system pin

printf(">>> Enter the pin number: ");

scanf("%d",&epass);

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

if(epass==apass){

while(1){

printf("################################################\n");

printf("------------- LPU COURIER SERVICES -------------\n");

printf("\n");

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

printf("\n");

printf(" ...Welcome to Dashboard...\n");

printf("\n");

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

int cmd; // cmd: command for action like: 1,2...8

printf("### Choose action:\n\n");

printf(">> 1 For Adding courier\n");

printf(">> 2 For Displaying courier\n");

printf(">> 3 For Updating delivery status\n");

printf(">> 4 For Tracking courier\n");

printf(">> 5 For Sorting(id) courier\n");

printf(">> 6 For Show delivered courier\n");

printf(">> 7 For Changing system pin\n");

printf(">> 8 For Exiting system\n");

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

printf(">>> Enter the command: ");

scanf("%d",&cmd);

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

switch(cmd){

case 1:

addcourier(); // calling function for adding courier

break;

case 2:

showcourier(); // calling function for displaying pending courier

break;

case 3:

updatecourier(); // calling function for updating delivery status

break;

case 4:

searchcourier(); // calling function for tracking courier

break;

case 5:

sortcourier(); // calling function for sorting courier based on asc. id

break;

case 6:

delcourier(); // calling function for displaying delivered courier

break;

case 7:

changepin(); // calling function for changing system pin

break;

case 8:

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

printf(" ...Thank You, Visit Again...\n");

printf(" Exited\n");

printf("#################################################");

exit(0);

default:

printf("\n");

printf("Invalid choice....\n");

printf("\n");

}

}

}

else{

printf("\n");

printf(" ! INVALID PIN, TRY AGAIN....");

printf("\n");

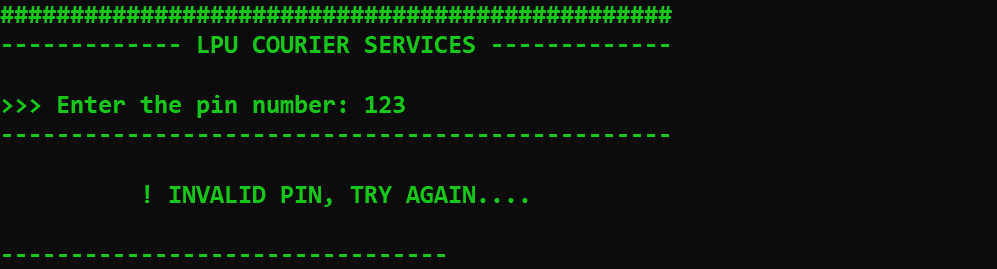
}

return 0;

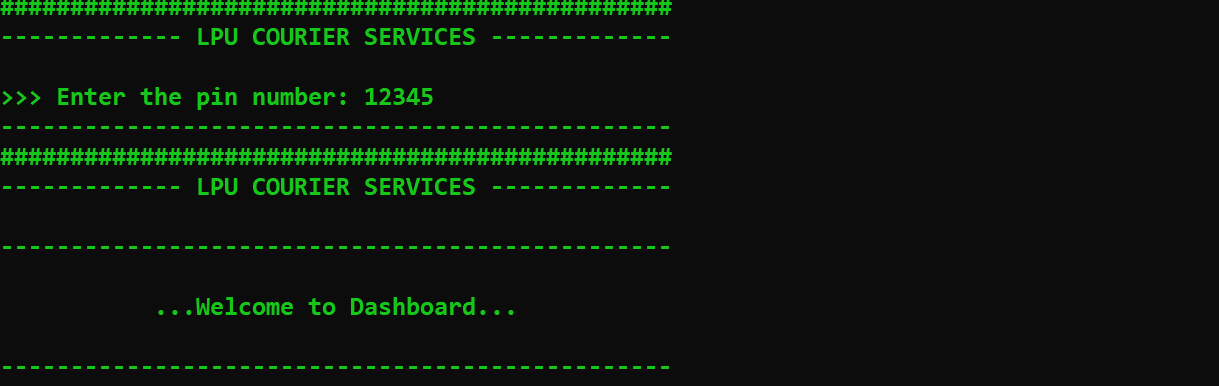
}

1. **RESULTS**

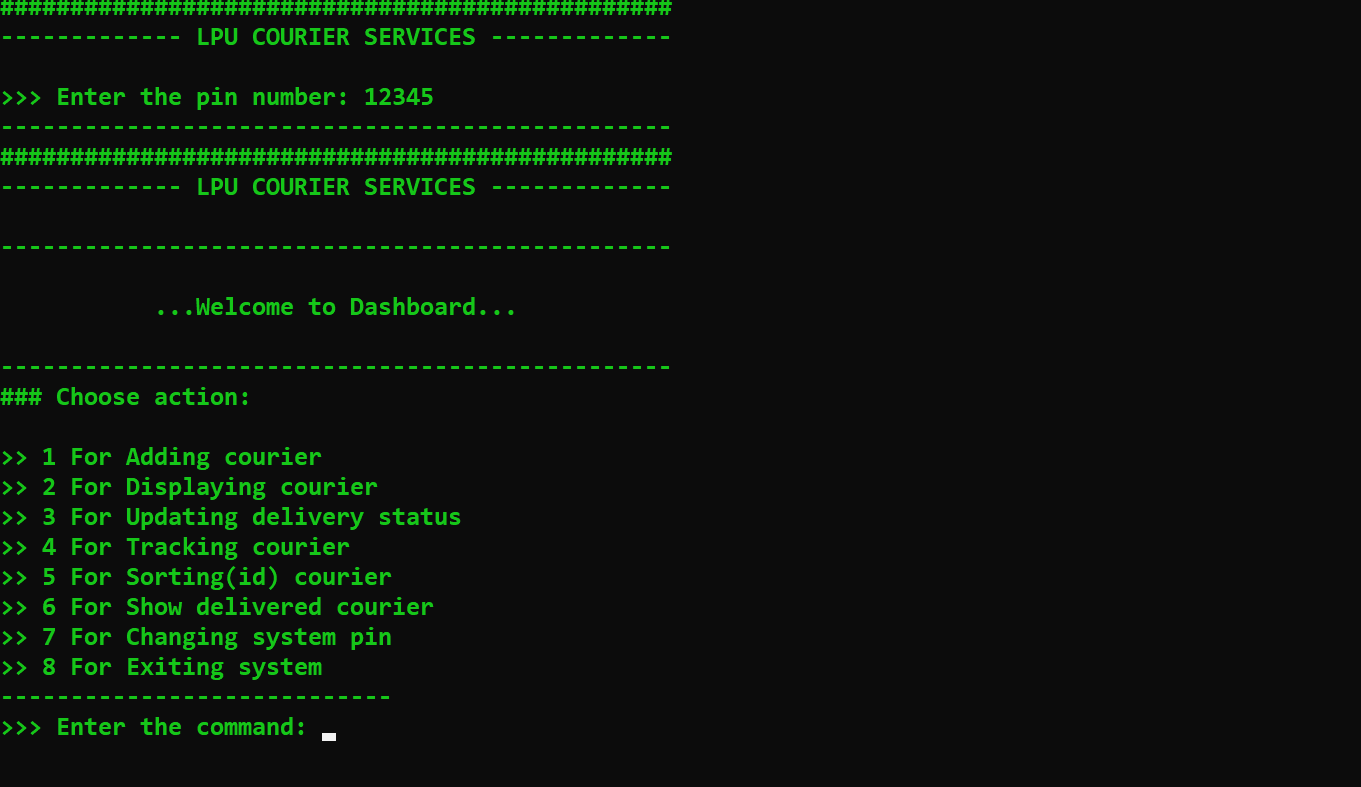
**4.1-Wrong login/pin input:**

****

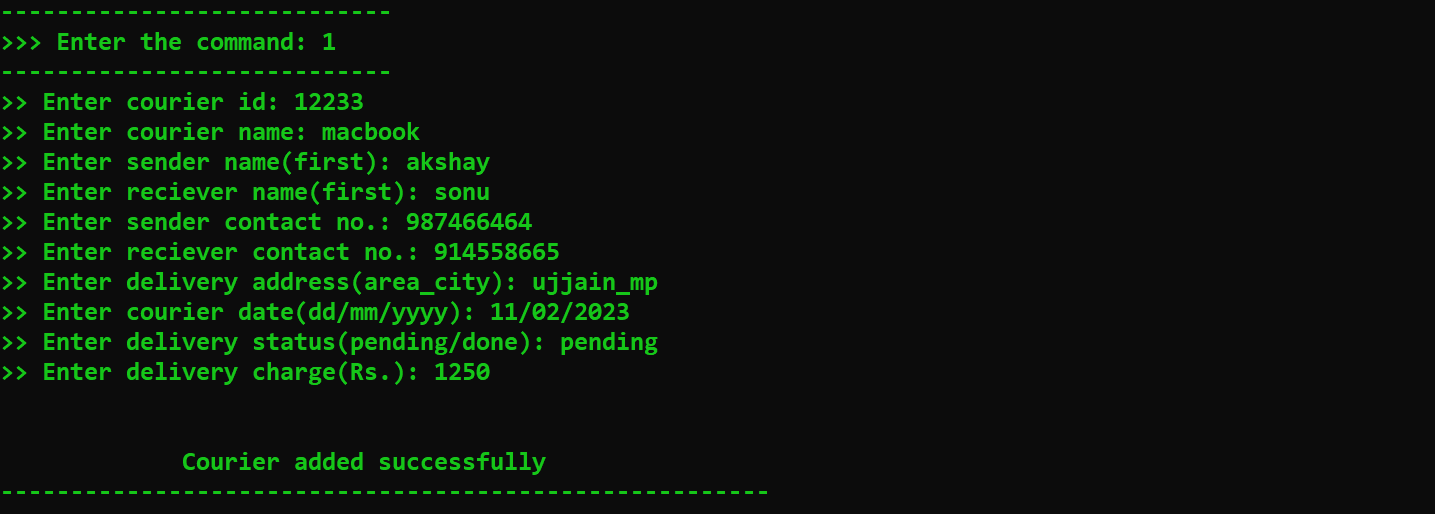
**4.2-Correct login/pin input:**



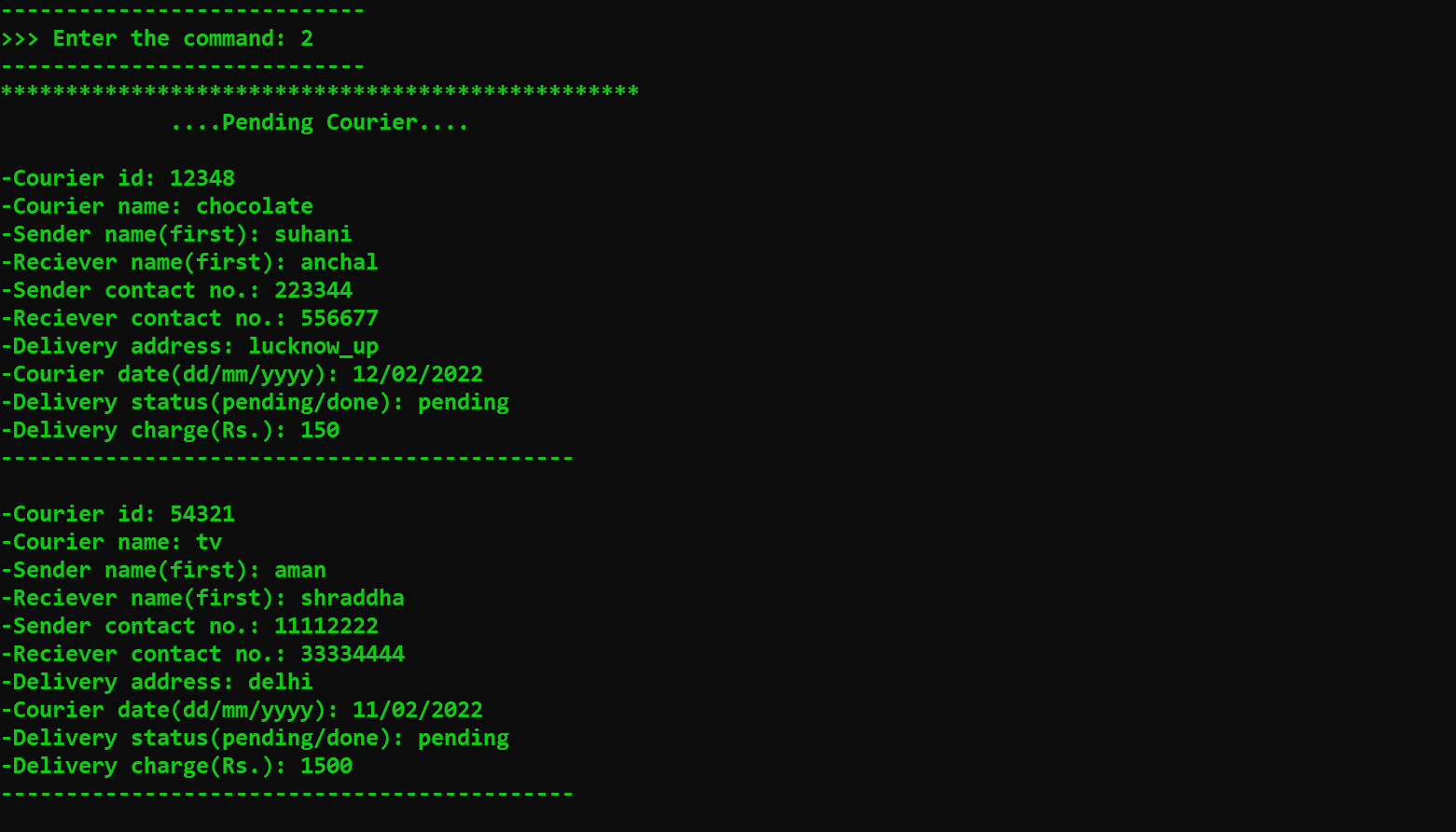
**4.3-Options for different actions:**

****

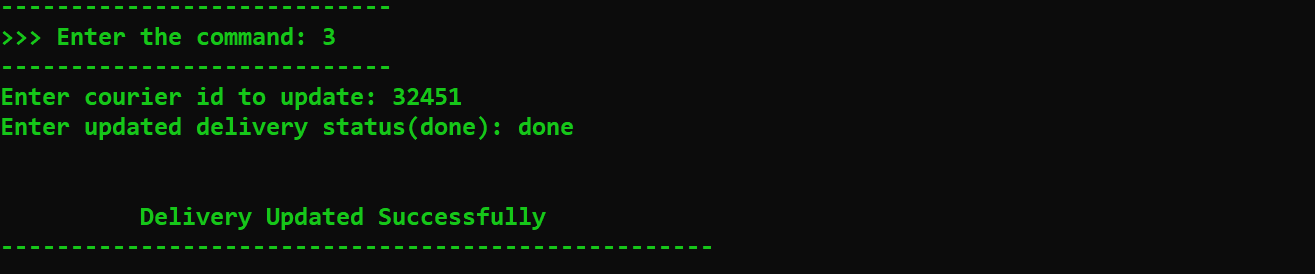
**4.4-Adding courier:**

****

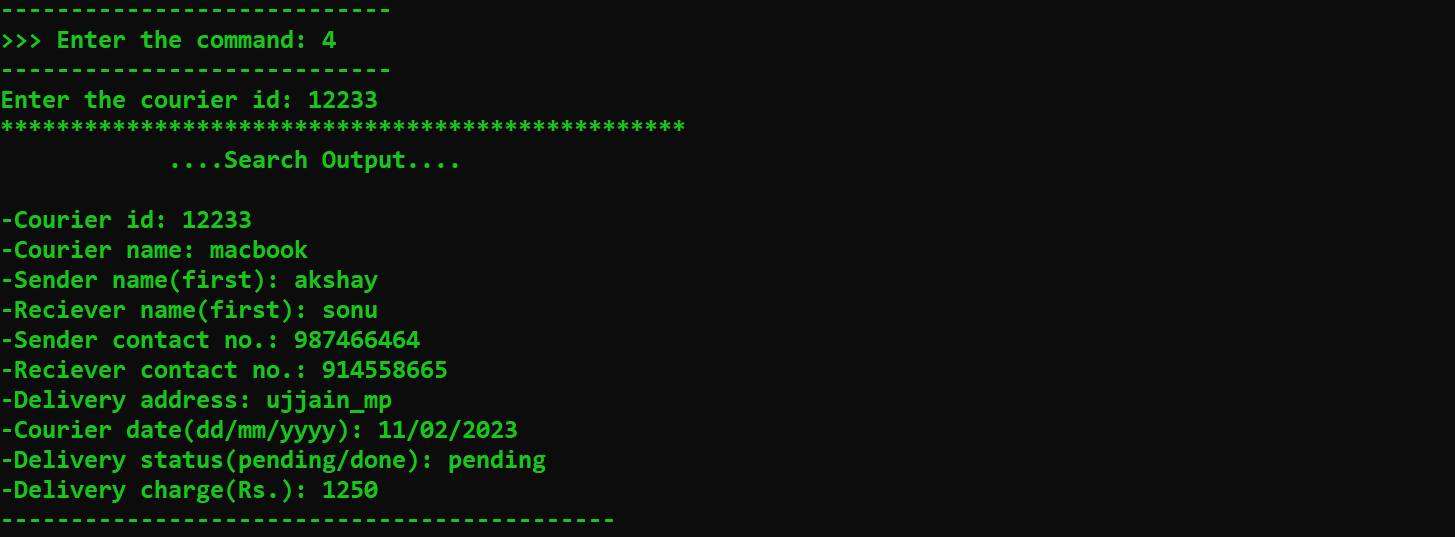
**4.5-Displaying courier:**

****

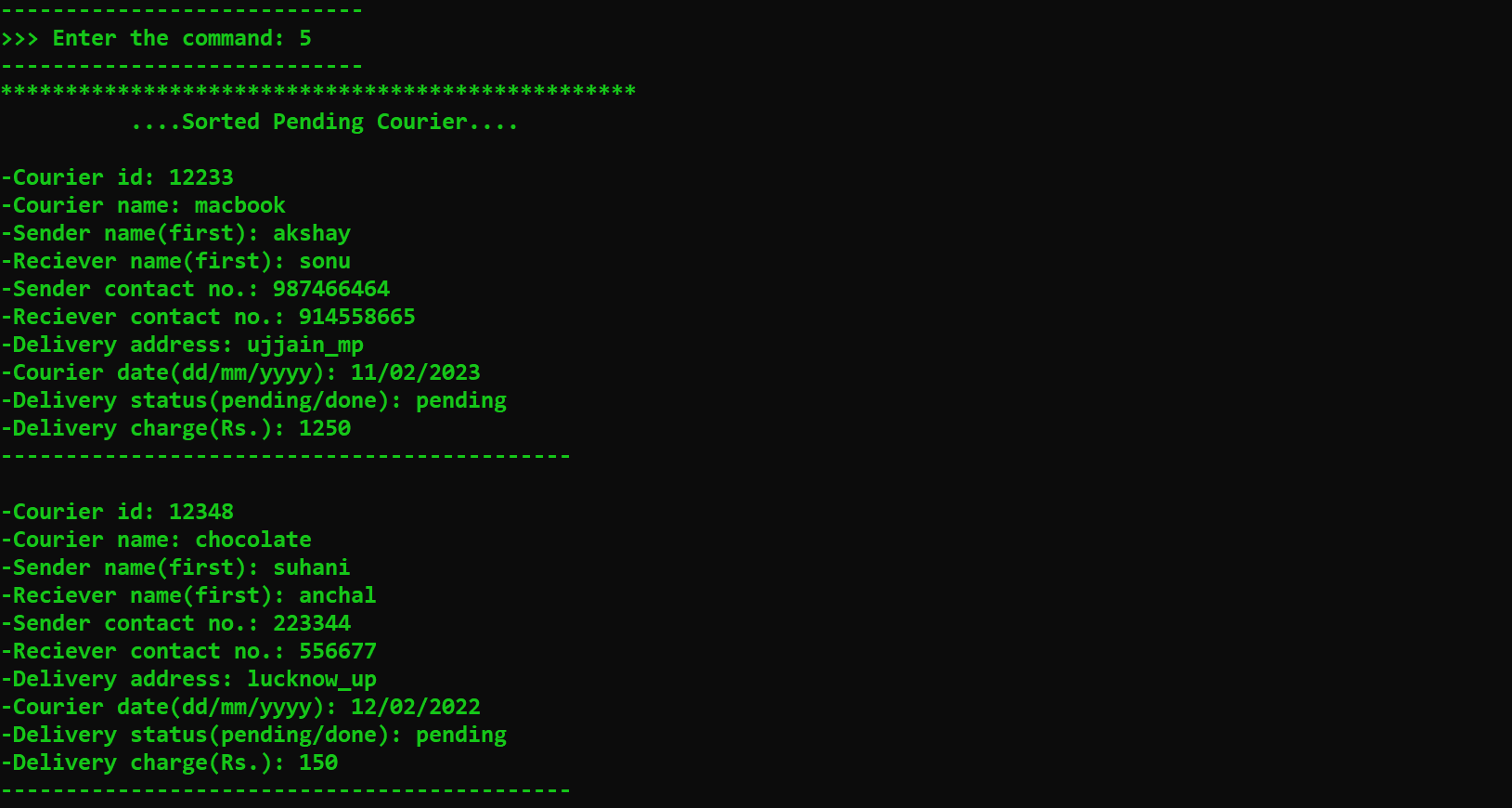
**4.6-Update delivery status:**



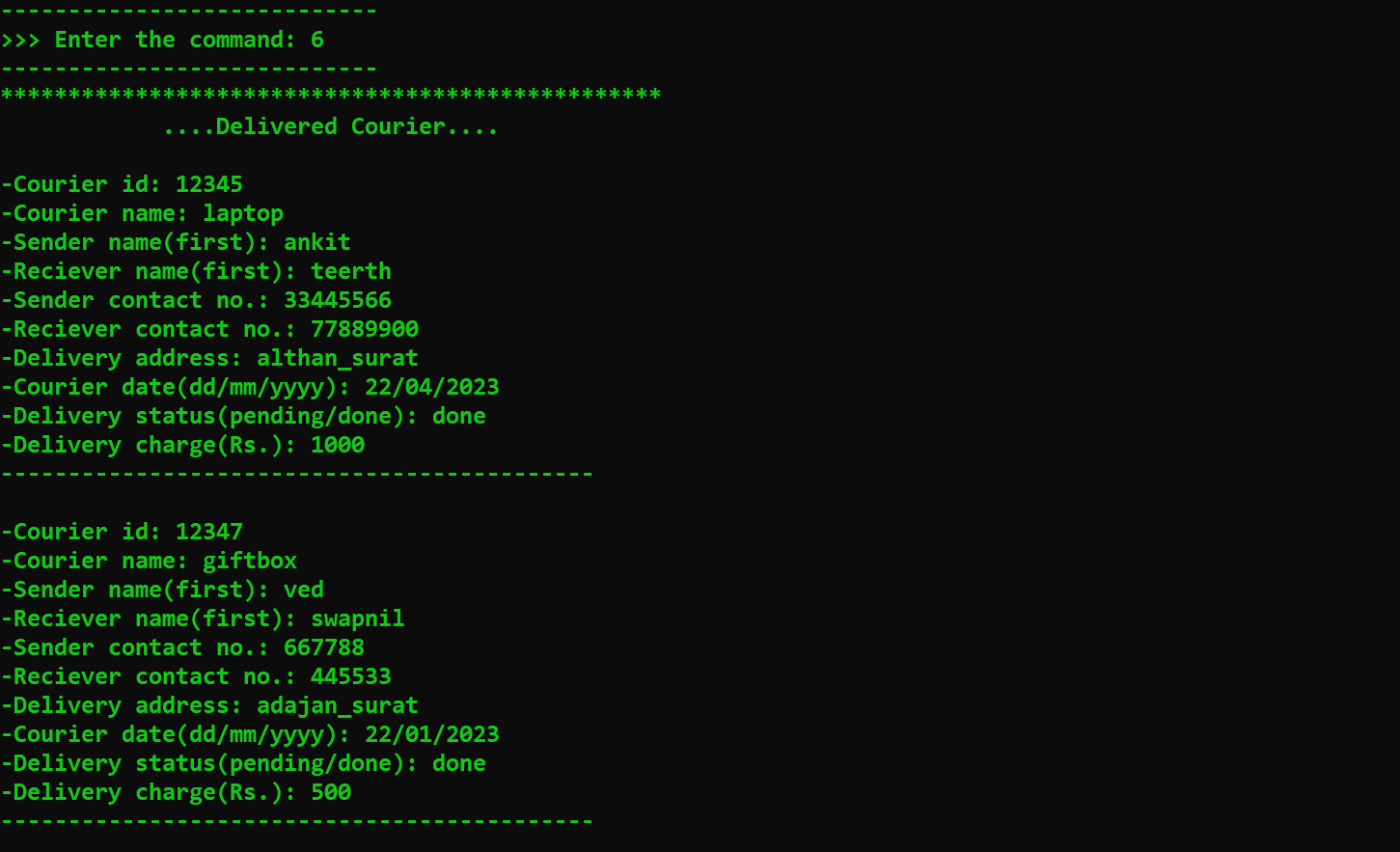
**4.7-Tracking courier:**

****

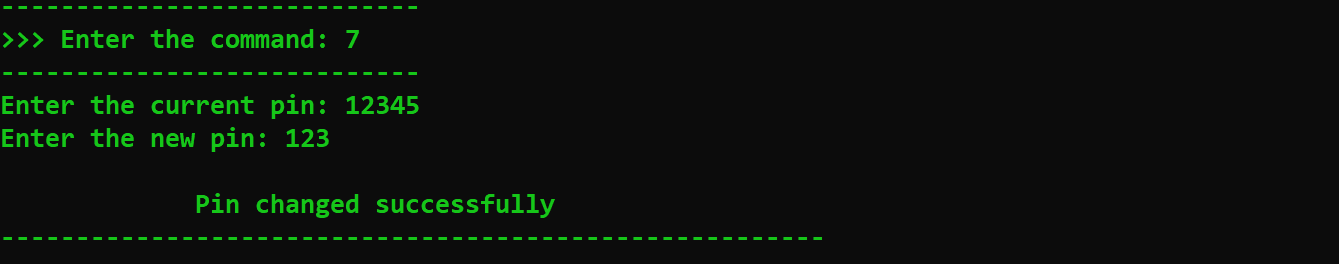
**4.8-Sorting courier(asc id):**



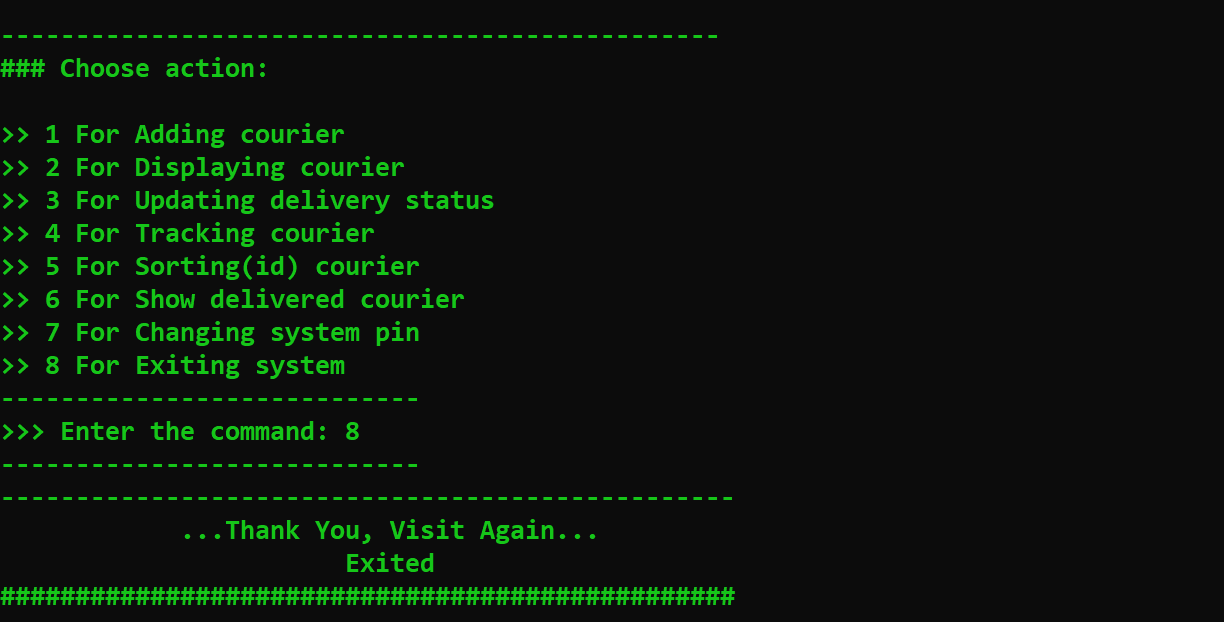
**4.9-Showing delivered courier:**

****

**4.10-Change system pin:**



**4.11-Exiting the system:**

****

**THANK YOU**

END OF PROJECT REPORT

\*\*\*\*\*\*\*\*