DATA STRUCTURE AND OBJECT ORIENTED PROGRAMMING LAB

FINAL PROJECT REPORT

GROUP MEMBERS:

SYED FASIH UL HUSSNAIN	(191029)
YASIR ZAHID	(191017)
OMER	(180847)

DEPARTMENT:

MECHATRONICS

CLASS:

BEMTS-F-19 – III ‘B’

INSTRUCTOR:

MA’AM SAMIA AZRAB

DATE:

1ST JAN 2021

FINAL PROJECT REPORT

OBJECTIVE:

This program is basically a CD Management Software which can be used in a CDs Store. It keeps the record of all the CDs present in the CDs Store and we can also add more records for the fresh arrivals. This project can search the CDs available in the store, by various modes.

PROJECT DISCRIPTION:

This C++ project is based on CD Management System. This program uses the concept of object oriented programming. Database is a collection of interrelated data to serve multiple applications. In this project we had used the knowledge of classes and linked list. In the CDs shop management system, we had created multiple functions which are explain as under:

CHECK IN CD:

The CD Management System contains a function of Check In. In this function there is detail related to the new CD came into Shop. This function helps us to look how many CDs came into the shop.

LIST OF CDs:

The Program we created contains the function in which we store the list of CDs present in shop. In this function we create the list of CDs that the CD stores owns. This function helps us to see the list of the CDs present in store by which it is easier to find any CD.

DETAILS OF PARTICULAR CD:

CD Management system also contains the function in which we store the details of particular CD. In this function we input the detail related to a specific CD present in store. This function helps us to find the CD according to the details given by the customer which will save our time.

PRINT LIST OF CDs:

CD Management software contains the function of printing the list of CDs. This function gets the information from List of CDs function and display on the screen. This will help us when a costumer asks to show the all CDs of the Store. By using this function, we print the list of CDs present in the store.

PARTICULAR CD IN STORE:

Our CD Management software also contains the function of Particular CD in store. This function will help the store manager to search the particular CD in the Store. Suppose a customer wants a particular CD and he ask for a specific CD this function helps a lot to locate the specific CD. The following functions are performed in the software.

MAINTAIN DATABASE:

This management software also maintains the database of costumer. This function will help the store manager to maintain the record of the costumers who rented the CDs or who have given back the CDs. It saves the shop owners time and memory to remind the Information related to the Costumers.

LIST OF CDs RENTED BY COSTUMERS:

Our software contains the function in which list of CDs rented by the costumer. This function helps the owner of the shop to maintain the record of the rented CDs by the costumer.

STORE COSTUMERS LIST:

The management software also contains the function in which list of costumers is stores. It helps the shop owner to made the list of its costumer who had taken the CDs on rent.

RETURN CD:

In the CDs Management Software there is an another function of returning the CD. This function shows the details related to the CDs which a costumer returns. It helps the CD shop manager to maintain the check and balance in the shop.

CLASSES:

The classes we used in the program are as follows:

- **CD**
- **CUSTOMER**

VARIABLES OF CLASS CD:

The variables of the class CD are:

- Int index,
- Int total copies,
- Int rented copies,
- Int available copies,
- Int no of stars,
- String name,
- String stars [10].

VARIABLES OF CLASS CUSTOMER:

The variables of the class CUSTOMER are**:**

- Int index;
- Int cds_rented;
- Int account_number;

- Int rented_cd_list [15];
- string name;

WORKING:

After the compilation of code, a main console is shown in which the message is displayed “welcome to the CD store Management:”. Then it will ask “HOW MANY CDS DETAILS WOULD YOU LIKE TO ENTER?" after the number of CDs it will ask that “how many costumers you want?” Afterwards entering the costumer’s and CDs list details it will ask you which information do u you want. Following menu is shown to you:

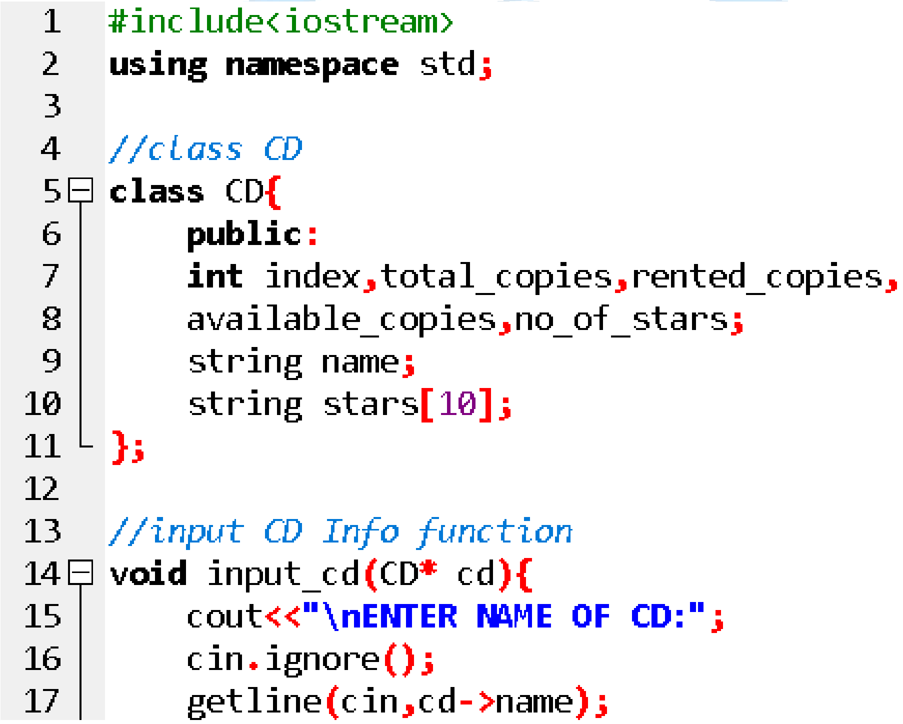
ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM

1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10. EXIT

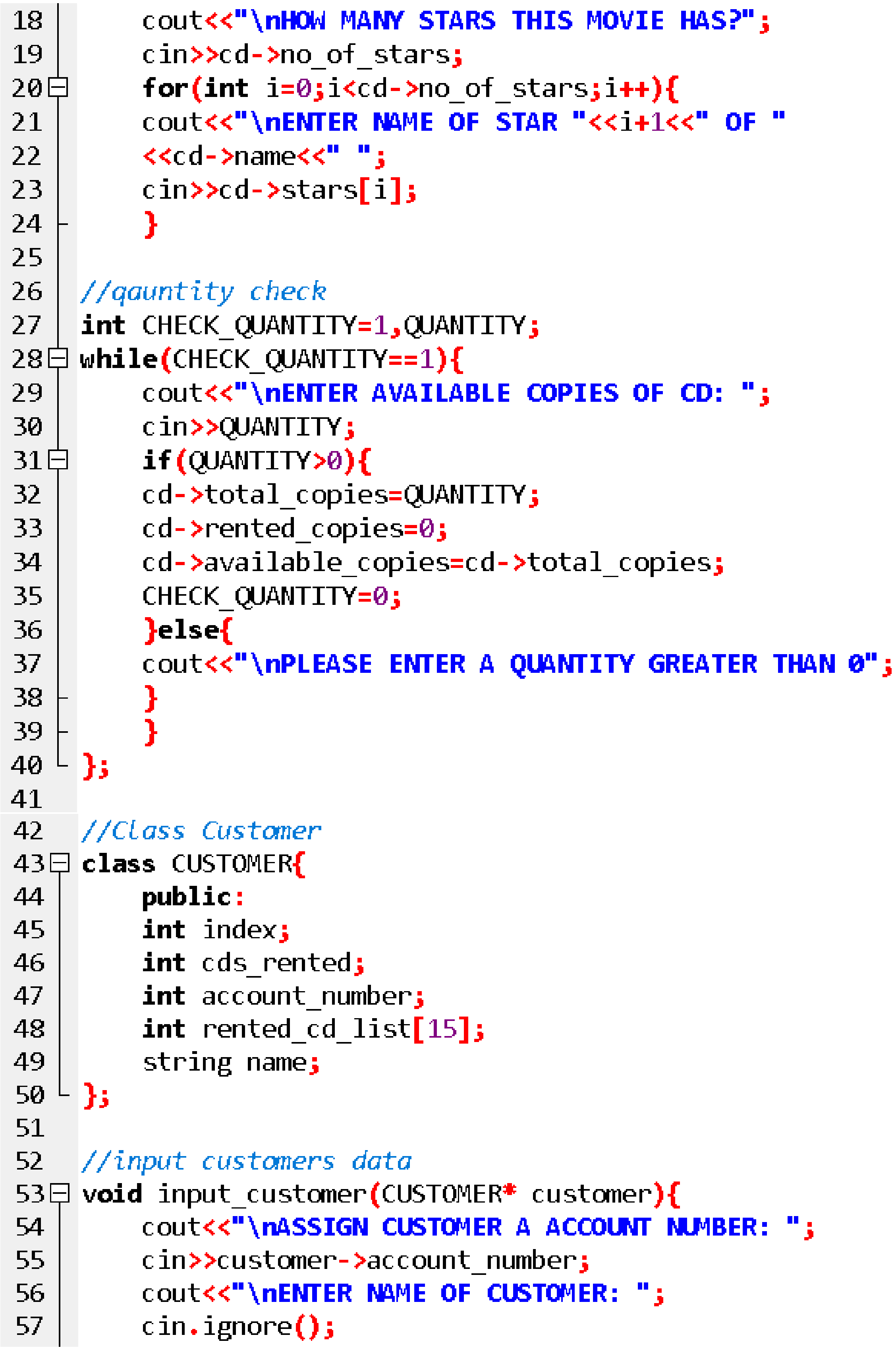
By choosing any of the above option you can perform any task on your screen.

CODE:

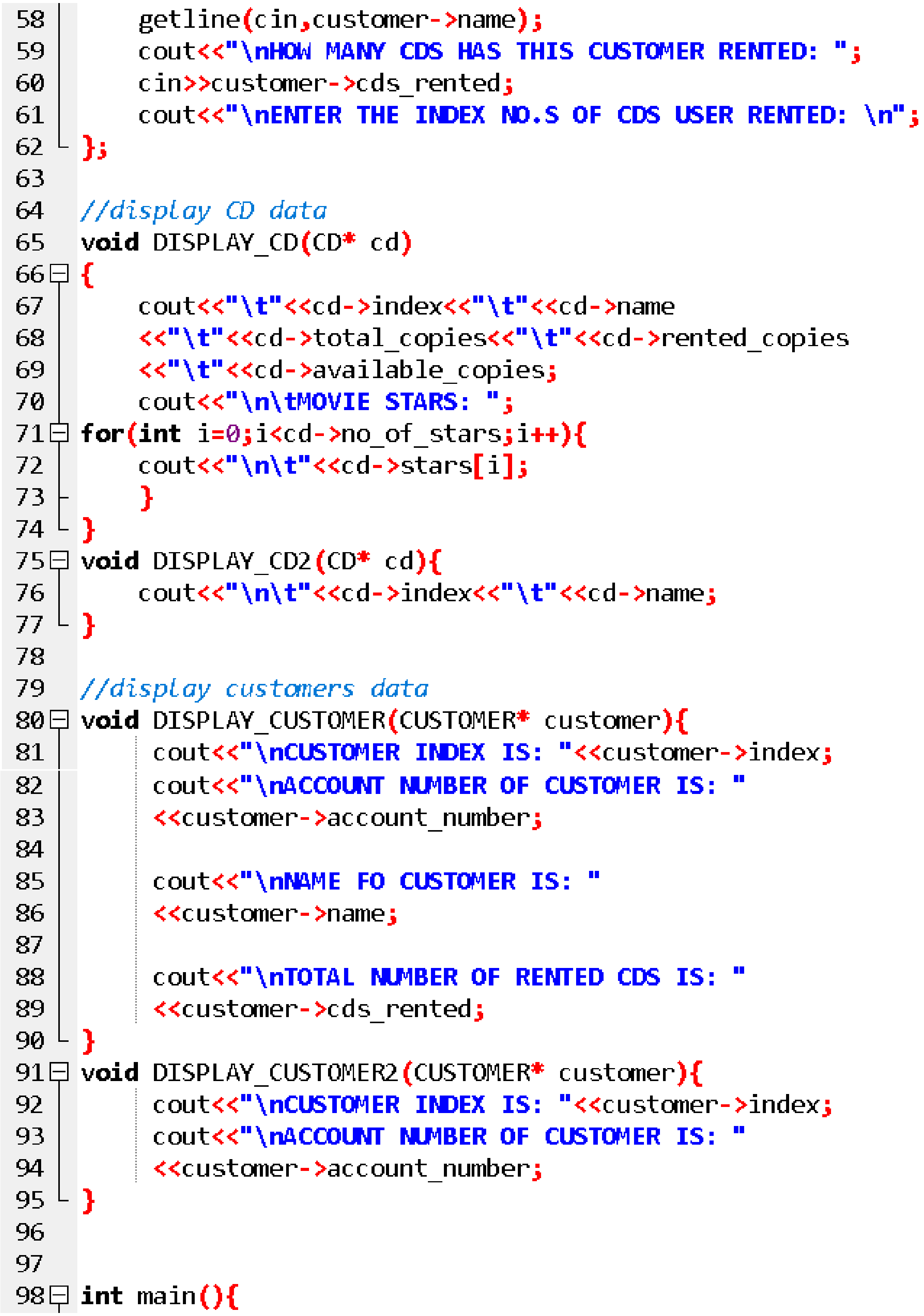
INPUT



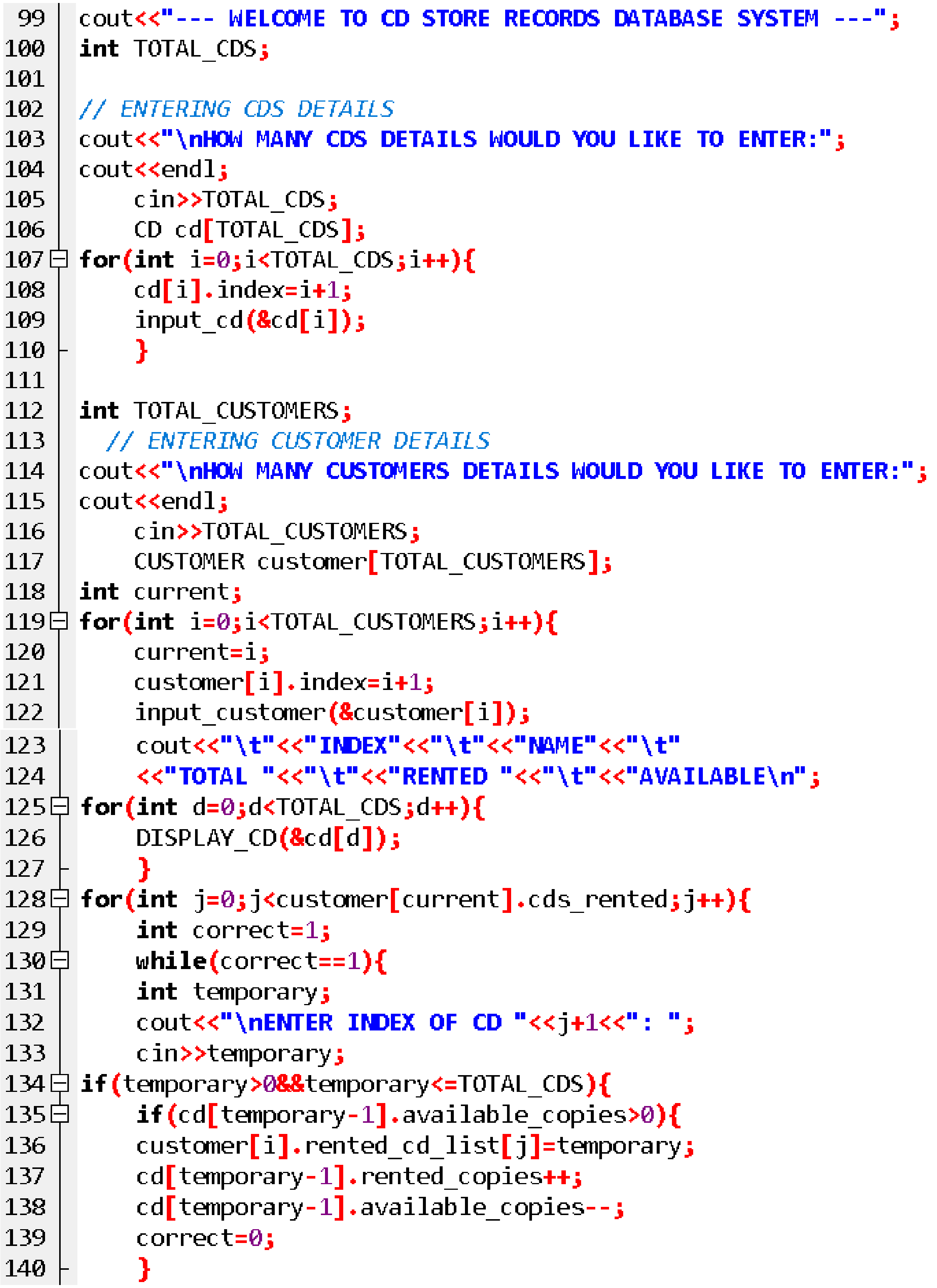
```
#include<iostream>
using namespace std;

//class CD
class CD{
    public:
    int index,total_copies,rented_copies,
    available_copies,no_of_stars;
    string name;
    string stars[10];
};

//input CD Info function
void input_cd(CD* cd){
    cout<<"\nENTER NAME OF CD:";
    cin.ignore();
    getline(cin,cd->name);
```



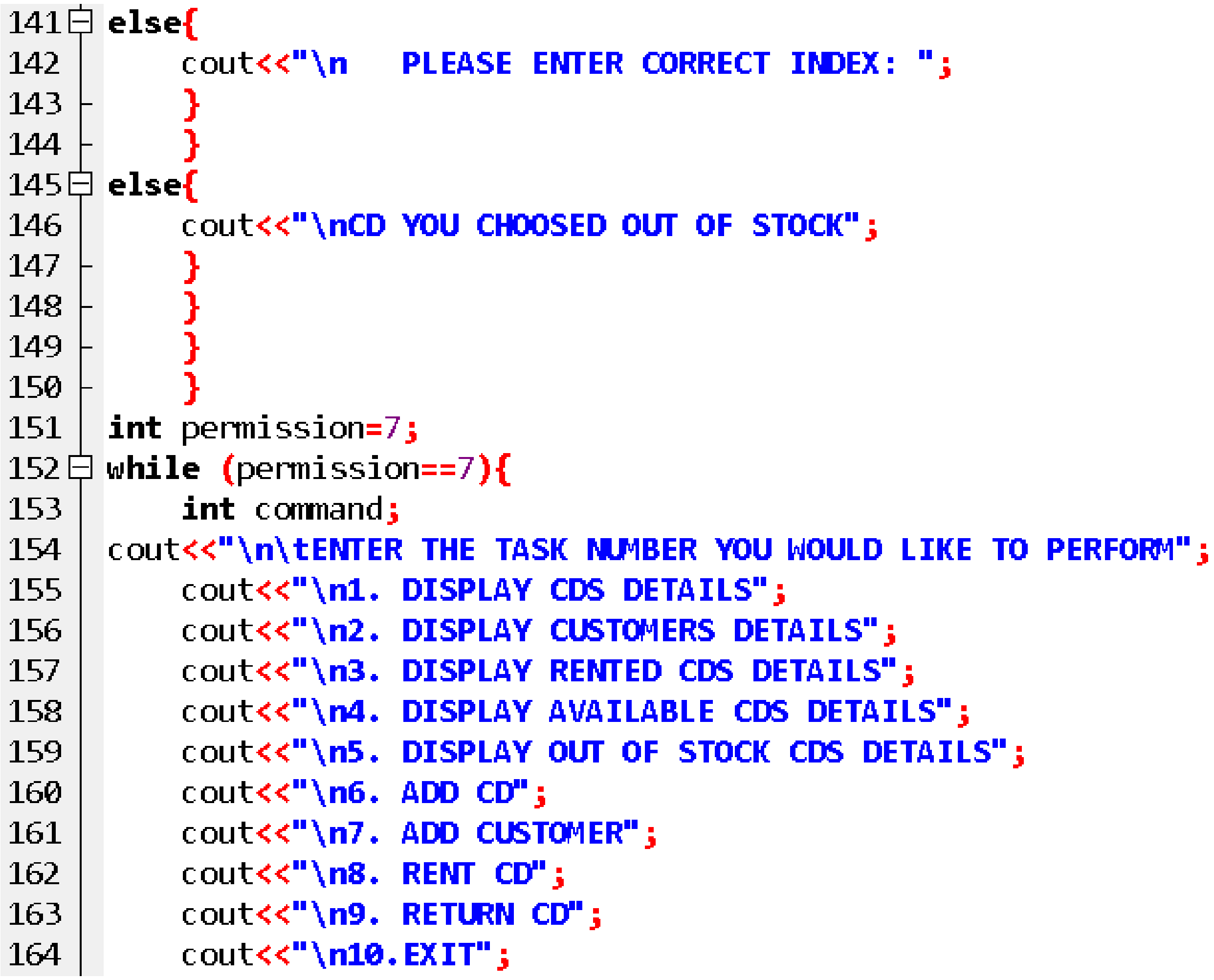
```
    cout<<"\nHOW MANY STARS THIS MOVIE HAS?";
    cin>>cd->no_of_stars;
    for(int i=0;i<cd->no_of_stars;i++){
    cout<<"\nENTER NAME OF STAR "<<i+1<<" OF "
    <<cd->name<<" ";
    cin>>cd->stars[i];
    }

//qauntity check
int CHECK_QUANTITY=1,QUANTITY;
while(CHECK_QUANTITY==1){
    cout<<"\nENTER AVAILABLE COPIES OF CD: ";
    cin>>QUANTITY;
    if(QUANTITY>0){
    cd->total_copies=QUANTITY;
    cd->rented_copies=0;
    cd->available_copies=cd->total_copies;
    CHECK_QUANTITY=0;
    }else{
    cout<<"\nPLEASE ENTER A QUANTITY GREATER THAN 0";
    }
    }
};

//Class Customer
class CUSTOMER{
    public:
    int index;
    int cds_rented;
    int account_number;
    int rented_cd_list[15];
    string name;
};

//input customers data
void input_customer(CUSTOMER* customer){
    cout<<"\nASSIGN CUSTOMER A ACCOUNT NUMBER: ";
    cin>>customer->account_number;
    cout<<"\nENTER NAME OF CUSTOMER: ";
    cin.ignore();
```



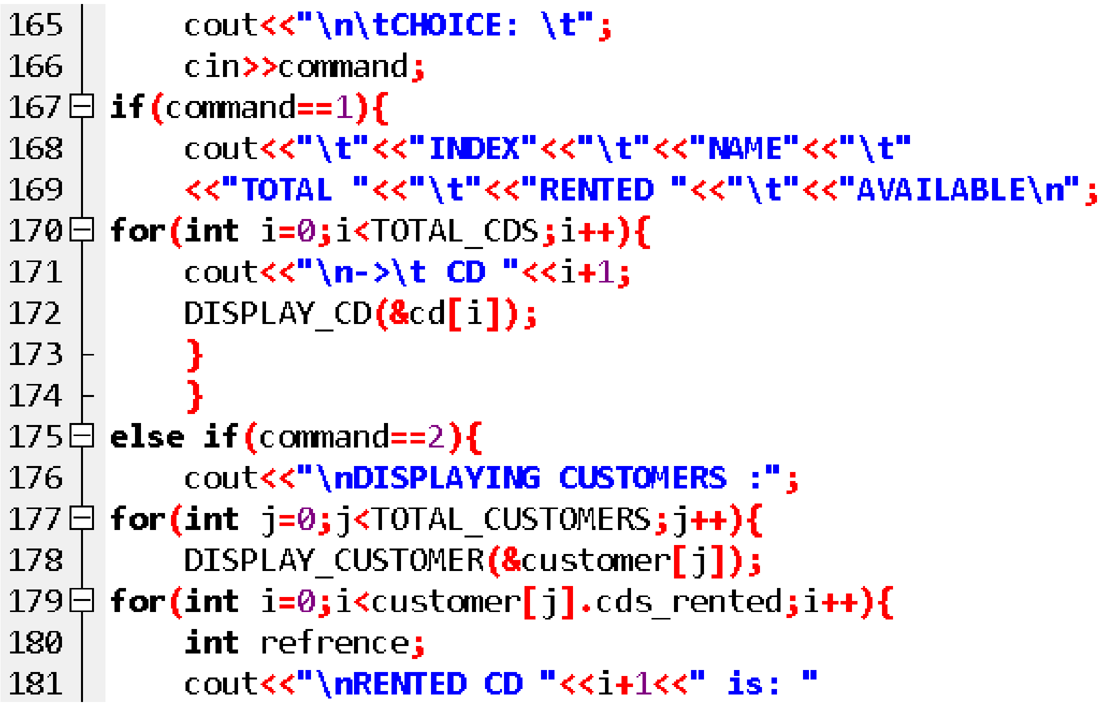
```cpp
    getline(cin,customer->name);
    cout<<"\nHOW MANY CDS HAS THIS CUSTOMER RENTED: ";
    cin>>customer->cds_rented;
    cout<<"\nENTER THE INDEX NO.S OF CDS USER RENTED: \n";
};

//display CD data
void DISPLAY_CD(CD* cd)
{
    cout<<"\t"<<cd->index<<"\t"<<cd->name
    <<"\t"<<cd->total_copies<<"\t"<<cd->rented_copies
    <<"\t"<<cd->available_copies;
    cout<<"\n\tMOVIE STARS: ";
for(int i=0;i<cd->no_of_stars;i++){
    cout<<"\n\t"<<cd->stars[i];
    }
}
void DISPLAY_CD2(CD* cd){
    cout<<"\n\t"<<cd->index<<"\t"<<cd->name;
}

//display customers data
void DISPLAY_CUSTOMER(CUSTOMER* customer){
     cout<<"\nCUSTOMER INDEX IS: "<<customer->index;
     cout<<"\nACCOUNT NUMBER OF CUSTOMER IS: "
     <<customer->account_number;

     cout<<"\nNAME FO CUSTOMER IS: "
     <<customer->name;

     cout<<"\nTOTAL NUMBER OF RENTED CDS IS: "
     <<customer->cds_rented;
}
void DISPLAY_CUSTOMER2(CUSTOMER* customer){
     cout<<"\nCUSTOMER INDEX IS: "<<customer->index;
     cout<<"\nACCOUNT NUMBER OF CUSTOMER IS: "
     <<customer->account_number;
}


int main(){
```



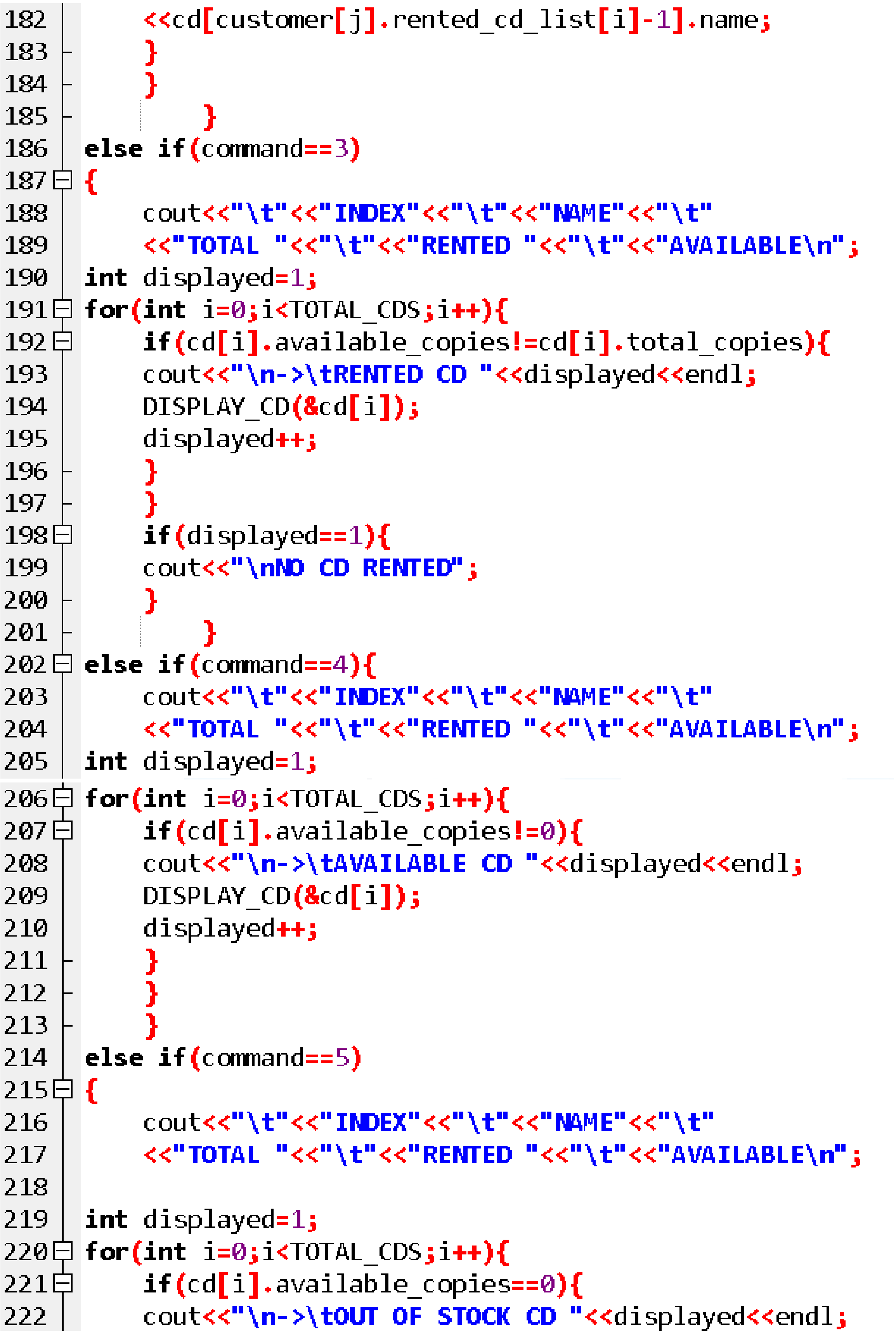
```
cout<<"--- WELCOME TO CD STORE RECORDS DATABASE SYSTEM ---";
int TOTAL_CDS;

// ENTERING CDS DETAILS
cout<<"\nHOW MANY CDS DETAILS WOULD YOU LIKE TO ENTER:";
cout<<endl;
    cin>>TOTAL_CDS;
    CD cd[TOTAL_CDS];
for(int i=0;i<TOTAL_CDS;i++){
    cd[i].index=i+1;
    input_cd(&cd[i]);
    }

int TOTAL_CUSTOMERS;
  // ENTERING CUSTOMER DETAILS
cout<<"\nHOW MANY CUSTOMERS DETAILS WOULD YOU LIKE TO ENTER:";
cout<<endl;
    cin>>TOTAL_CUSTOMERS;
    CUSTOMER customer[TOTAL_CUSTOMERS];
int current;
for(int i=0;i<TOTAL_CUSTOMERS;i++){
    current=i;
    customer[i].index=i+1;
    input_customer(&customer[i]);
    cout<<"\t"<<"INDEX"<<"\t"<<"NAME"<<"\t"
    <<"TOTAL "<<"\t"<<"RENTED "<<"\t"<<"AVAILABLE\n";
for(int d=0;d<TOTAL_CDS;d++){
    DISPLAY_CD(&cd[d]);
    }
for(int j=0;j<customer[current].cds_rented;j++){
    int correct=1;
    while(correct==1){
    int temporary;
    cout<<"\nENTER INDEX OF CD "<<j+1<<": ";
    cin>>temporary;
if(temporary>0&&temporary<=TOTAL_CDS){
    if(cd[temporary-1].available_copies>0){
    customer[i].rented_cd_list[j]=temporary;
    cd[temporary-1].rented_copies++;
    cd[temporary-1].available_copies--;
    correct=0;
    }
```



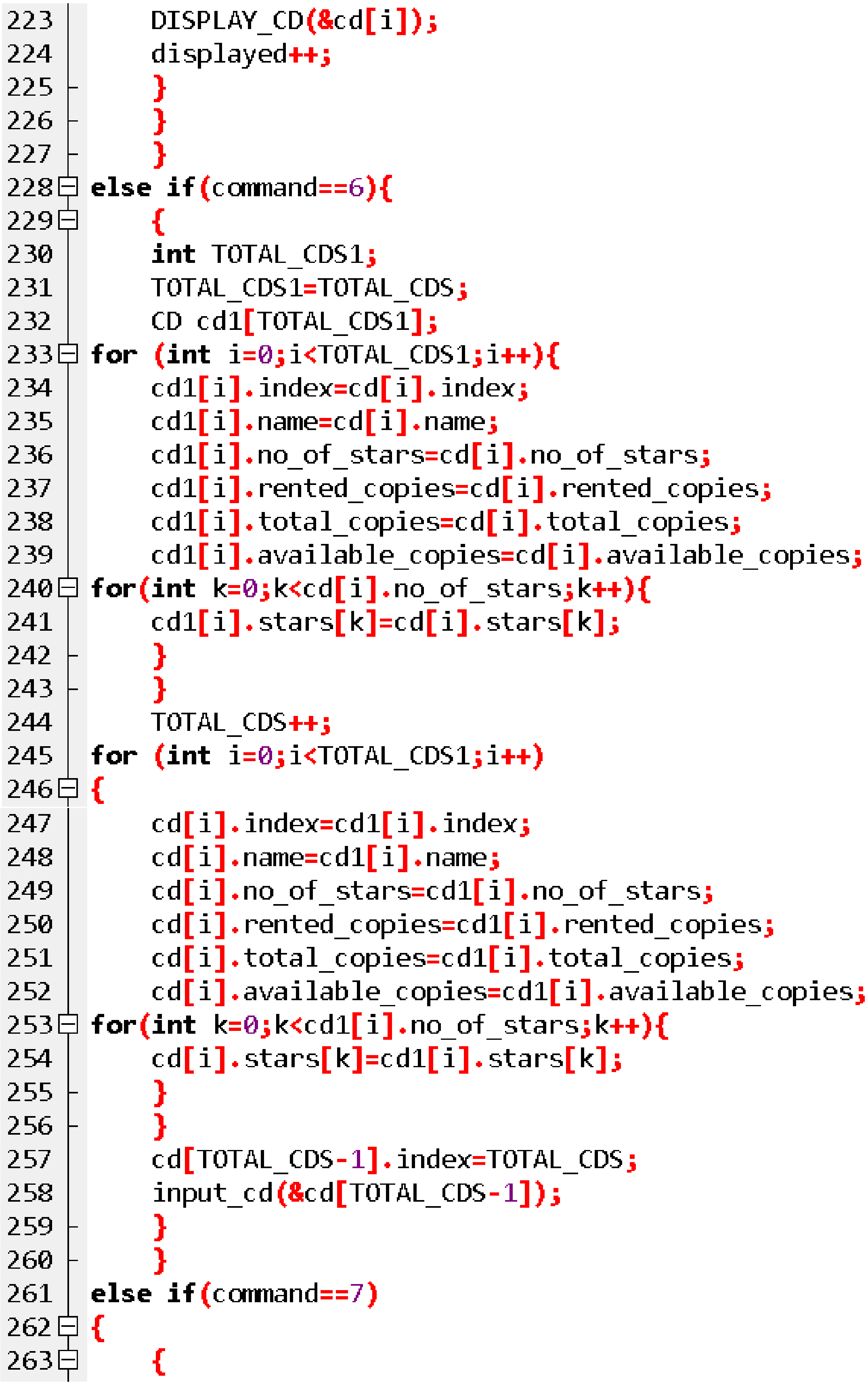
```
else{
    cout<<"\n   PLEASE ENTER CORRECT INDEX: ";
    }
    }
else{
    cout<<"\nCD YOU CHOOSED OUT OF STOCK";
    }
    }
    }
    }
int permission=7;
while (permission==7){
    int command;
cout<<"\n\tENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM";
    cout<<"\n1. DISPLAY CDS DETAILS";
    cout<<"\n2. DISPLAY CUSTOMERS DETAILS";
    cout<<"\n3. DISPLAY RENTED CDS DETAILS";
    cout<<"\n4. DISPLAY AVAILABLE CDS DETAILS";
    cout<<"\n5. DISPLAY OUT OF STOCK CDS DETAILS";
    cout<<"\n6. ADD CD";
    cout<<"\n7. ADD CUSTOMER";
    cout<<"\n8. RENT CD";
    cout<<"\n9. RETURN CD";
    cout<<"\n10.EXIT";
```



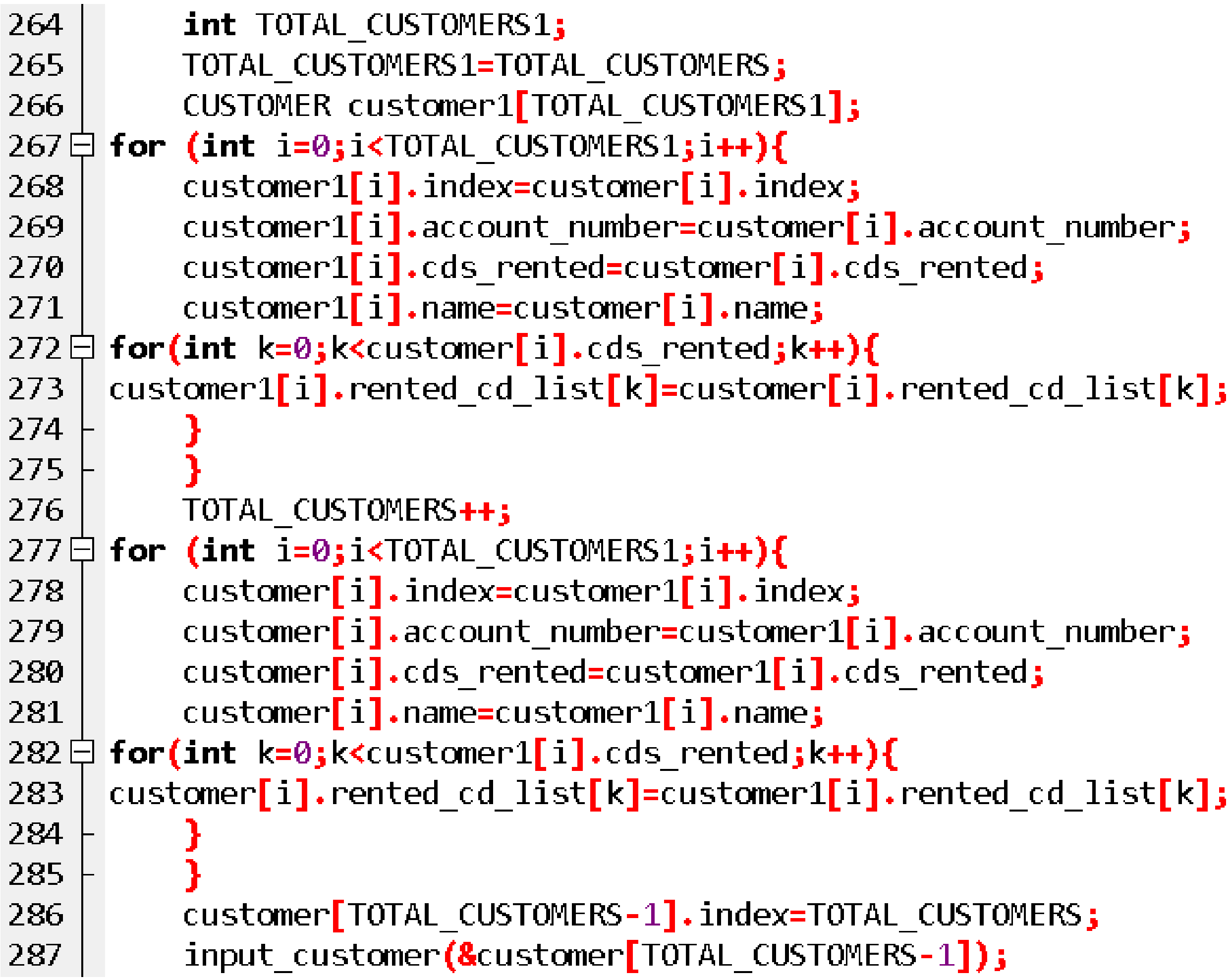
```
    cout<<"\n\tCHOICE: \t";
    cin>>command;
if(command==1){
    cout<<"\t"<<"INDEX"<<"\t"<<"NAME"<<"\t"
    <<"TOTAL "<<"\t"<<"RENTED "<<"\t"<<"AVAILABLE\n";
for(int i=0;i<TOTAL_CDS;i++){
    cout<<"\n->\t CD "<<i+1;
    DISPLAY_CD(&cd[i]);
    }
    }
else if(command==2){
    cout<<"\nDISPLAYING CUSTOMERS :";
for(int j=0;j<TOTAL_CUSTOMERS;j++){
    DISPLAY_CUSTOMER(&customer[j]);
for(int i=0;i<customer[j].cds_rented;i++){
    int refrence;
    cout<<"\nRENTED CD "<<i+1<<" is: "
```



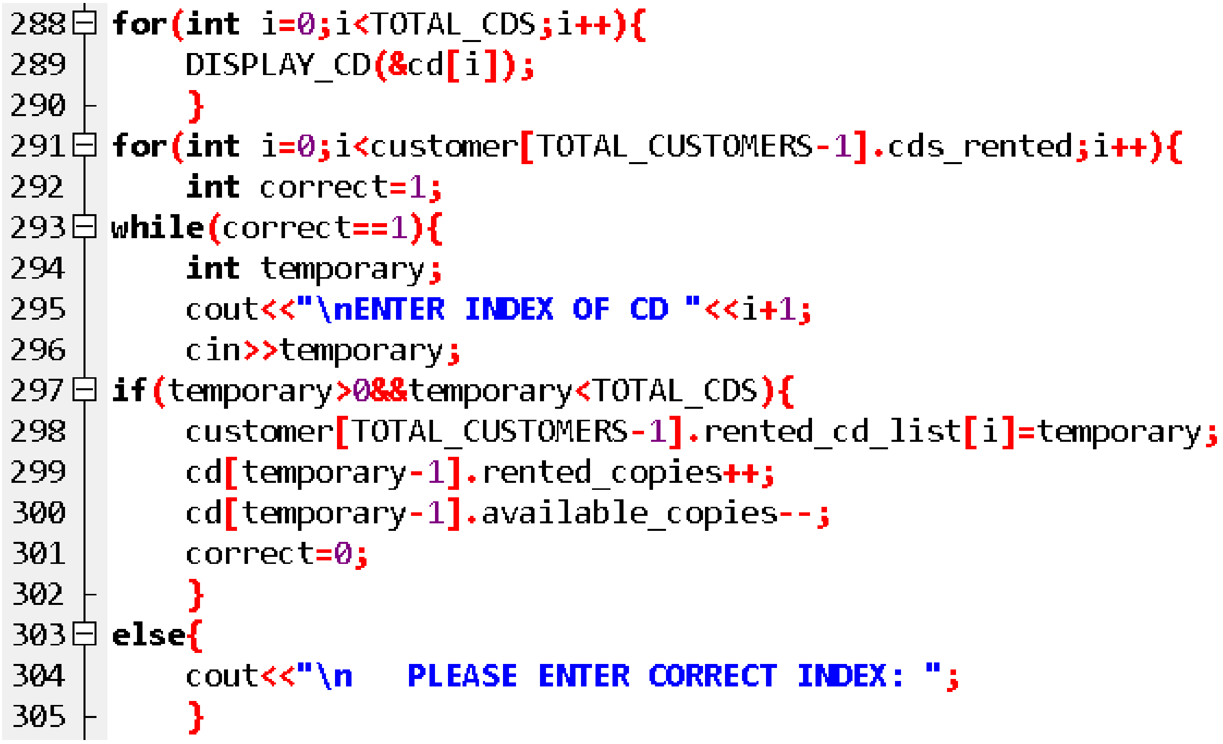
```
    <<cd[customer[j].rented_cd_list[i]-1].name;
    }
    }
        }
else if(command==3)
{
    cout<<"\t"<<"INDEX"<<"\t"<<"NAME"<<"\t"
    <<"TOTAL "<<"\t"<<"RENTED "<<"\t"<<"AVAILABLE\n";
int displayed=1;
for(int i=0;i<TOTAL_CDS;i++){
    if(cd[i].available_copies!=cd[i].total_copies){
    cout<<"\n->\tRENTED CD "<<displayed<<endl;
    DISPLAY_CD(&cd[i]);
    displayed++;
    }
    }
    if(displayed==1){
    cout<<"\nNO CD RENTED";
    }
        }
else if(command==4){
    cout<<"\t"<<"INDEX"<<"\t"<<"NAME"<<"\t"
    <<"TOTAL "<<"\t"<<"RENTED "<<"\t"<<"AVAILABLE\n";
int displayed=1;
for(int i=0;i<TOTAL_CDS;i++){
    if(cd[i].available_copies!=0){
    cout<<"\n->\tAVAILABLE CD "<<displayed<<endl;
    DISPLAY_CD(&cd[i]);
    displayed++;
    }
    }
    }
else if(command==5)
{
    cout<<"\t"<<"INDEX"<<"\t"<<"NAME"<<"\t"
    <<"TOTAL "<<"\t"<<"RENTED "<<"\t"<<"AVAILABLE\n";

int displayed=1;
for(int i=0;i<TOTAL_CDS;i++){
    if(cd[i].available_copies==0){
    cout<<"\n->\tOUT OF STOCK CD "<<displayed<<endl;
```



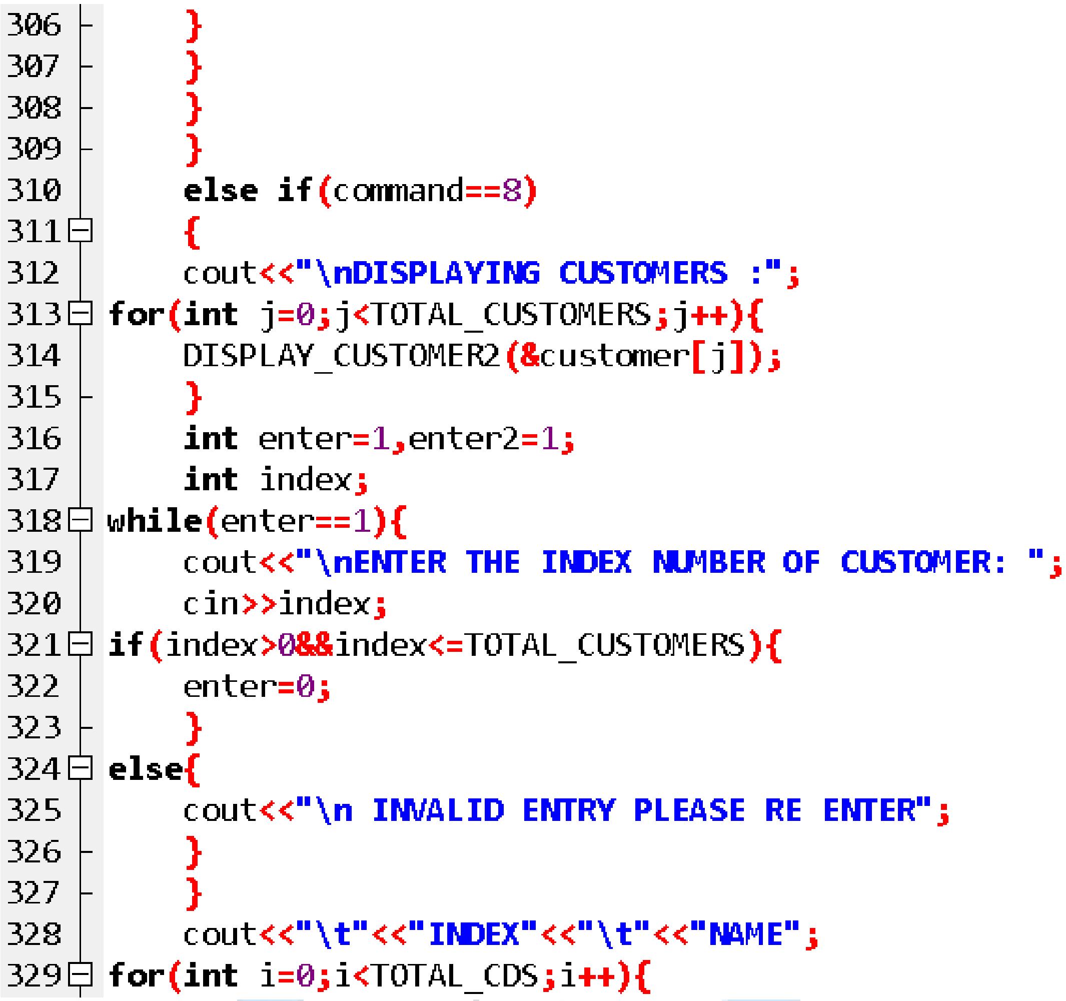
```
    DISPLAY_CD(&cd[i]);
    displayed++;
    }
    }
    }
else if(command==6){
    {
    int TOTAL_CDS1;
    TOTAL_CDS1=TOTAL_CDS;
    CD cd1[TOTAL_CDS1];
for (int i=0;i<TOTAL_CDS1;i++){
    cd1[i].index=cd[i].index;
    cd1[i].name=cd[i].name;
    cd1[i].no_of_stars=cd[i].no_of_stars;
    cd1[i].rented_copies=cd[i].rented_copies;
    cd1[i].total_copies=cd[i].total_copies;
    cd1[i].available_copies=cd[i].available_copies;
for(int k=0;k<cd[i].no_of_stars;k++){
    cd1[i].stars[k]=cd[i].stars[k];
    }
    }
    TOTAL_CDS++;
for (int i=0;i<TOTAL_CDS1;i++)
{
    cd[i].index=cd1[i].index;
    cd[i].name=cd1[i].name;
    cd[i].no_of_stars=cd1[i].no_of_stars;
    cd[i].rented_copies=cd1[i].rented_copies;
    cd[i].total_copies=cd1[i].total_copies;
    cd[i].available_copies=cd1[i].available_copies;
for(int k=0;k<cd1[i].no_of_stars;k++){
    cd[i].stars[k]=cd1[i].stars[k];
    }
    }
    cd[TOTAL_CDS-1].index=TOTAL_CDS;
    input_cd(&cd[TOTAL_CDS-1]);
    }
    }
else if(command==7)
{
    {
```



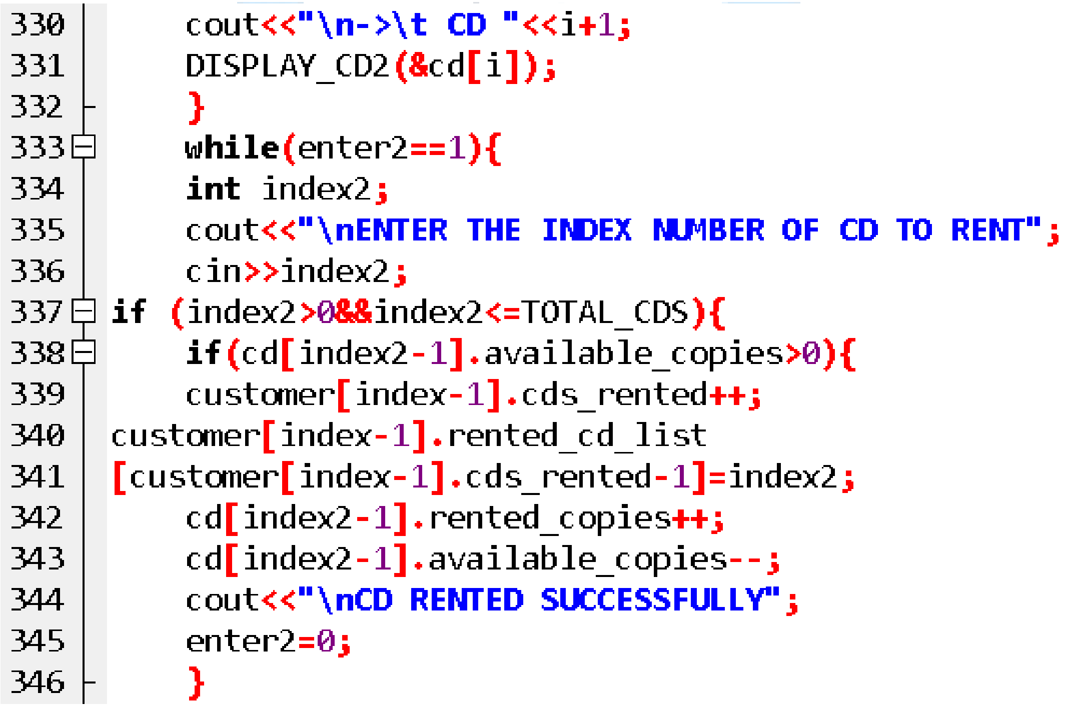
```
    int TOTAL_CUSTOMERS1;
    TOTAL_CUSTOMERS1=TOTAL_CUSTOMERS;
    CUSTOMER customer1[TOTAL_CUSTOMERS1];
for (int i=0;i<TOTAL_CUSTOMERS1;i++){
    customer1[i].index=customer[i].index;
    customer1[i].account_number=customer[i].account_number;
    customer1[i].cds_rented=customer[i].cds_rented;
    customer1[i].name=customer[i].name;
for(int k=0;k<customer[i].cds_rented;k++){
customer1[i].rented_cd_list[k]=customer[i].rented_cd_list[k];
    }
    }
    TOTAL_CUSTOMERS++;
for (int i=0;i<TOTAL_CUSTOMERS1;i++){
    customer[i].index=customer1[i].index;
    customer[i].account_number=customer1[i].account_number;
    customer[i].cds_rented=customer1[i].cds_rented;
    customer[i].name=customer1[i].name;
for(int k=0;k<customer1[i].cds_rented;k++){
customer[i].rented_cd_list[k]=customer1[i].rented_cd_list[k];
    }
    }
    customer[TOTAL_CUSTOMERS-1].index=TOTAL_CUSTOMERS;
    input_customer(&customer[TOTAL_CUSTOMERS-1]);
```



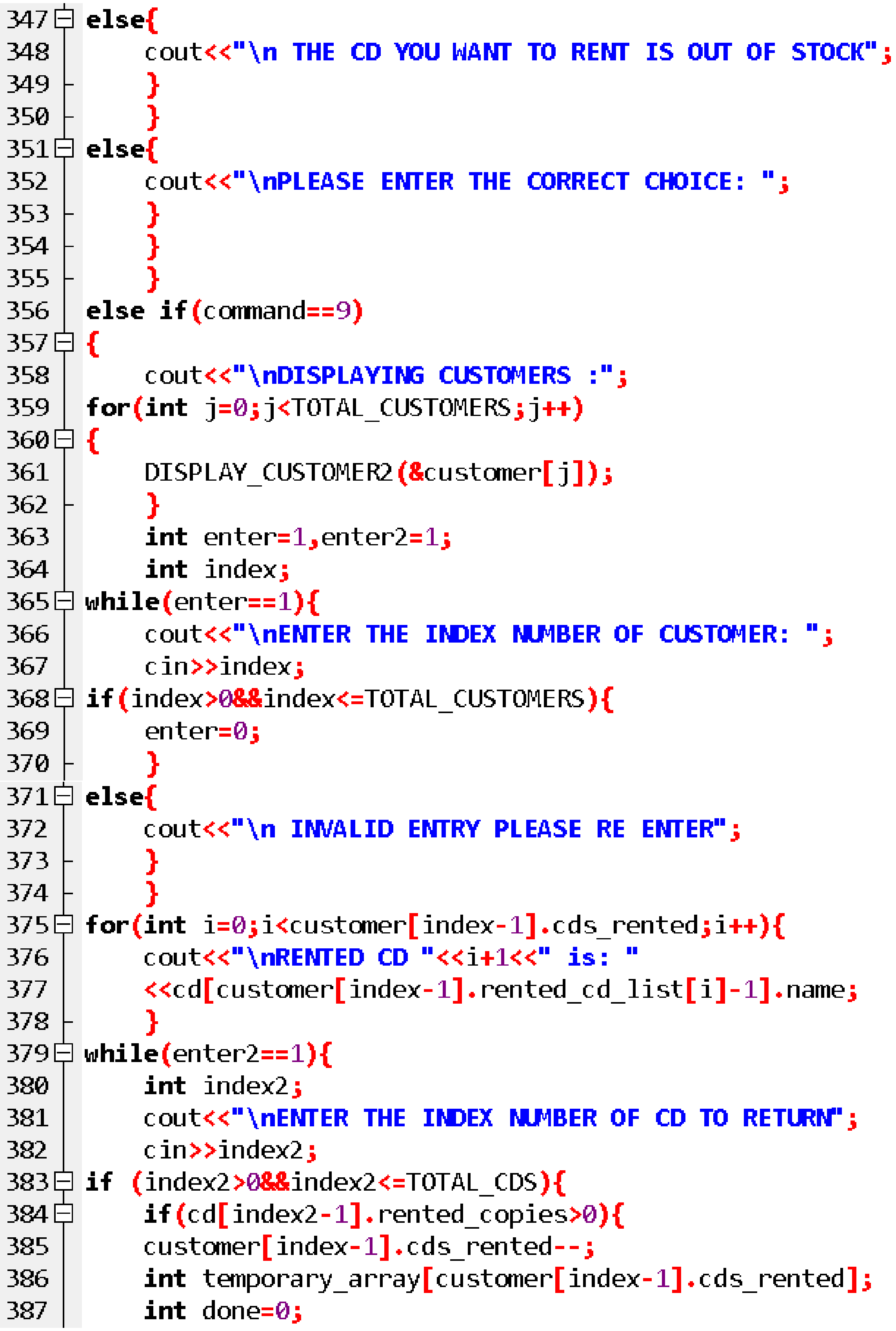
```
for(int i=0;i<TOTAL_CDS;i++){
    DISPLAY_CD(&cd[i]);
    }
for(int i=0;i<customer[TOTAL_CUSTOMERS-1].cds_rented;i++){
    int correct=1;
while(correct==1){
    int temporary;
    cout<<"\nENTER INDEX OF CD "<<i+1;
    cin>>temporary;
if(temporary>0&&temporary<TOTAL_CDS){
    customer[TOTAL_CUSTOMERS-1].rented_cd_list[i]=temporary;
    cd[temporary-1].rented_copies++;
    cd[temporary-1].available_copies--;
    correct=0;
    }
else{
    cout<<"\n   PLEASE ENTER CORRECT INDEX: ";
    }
```



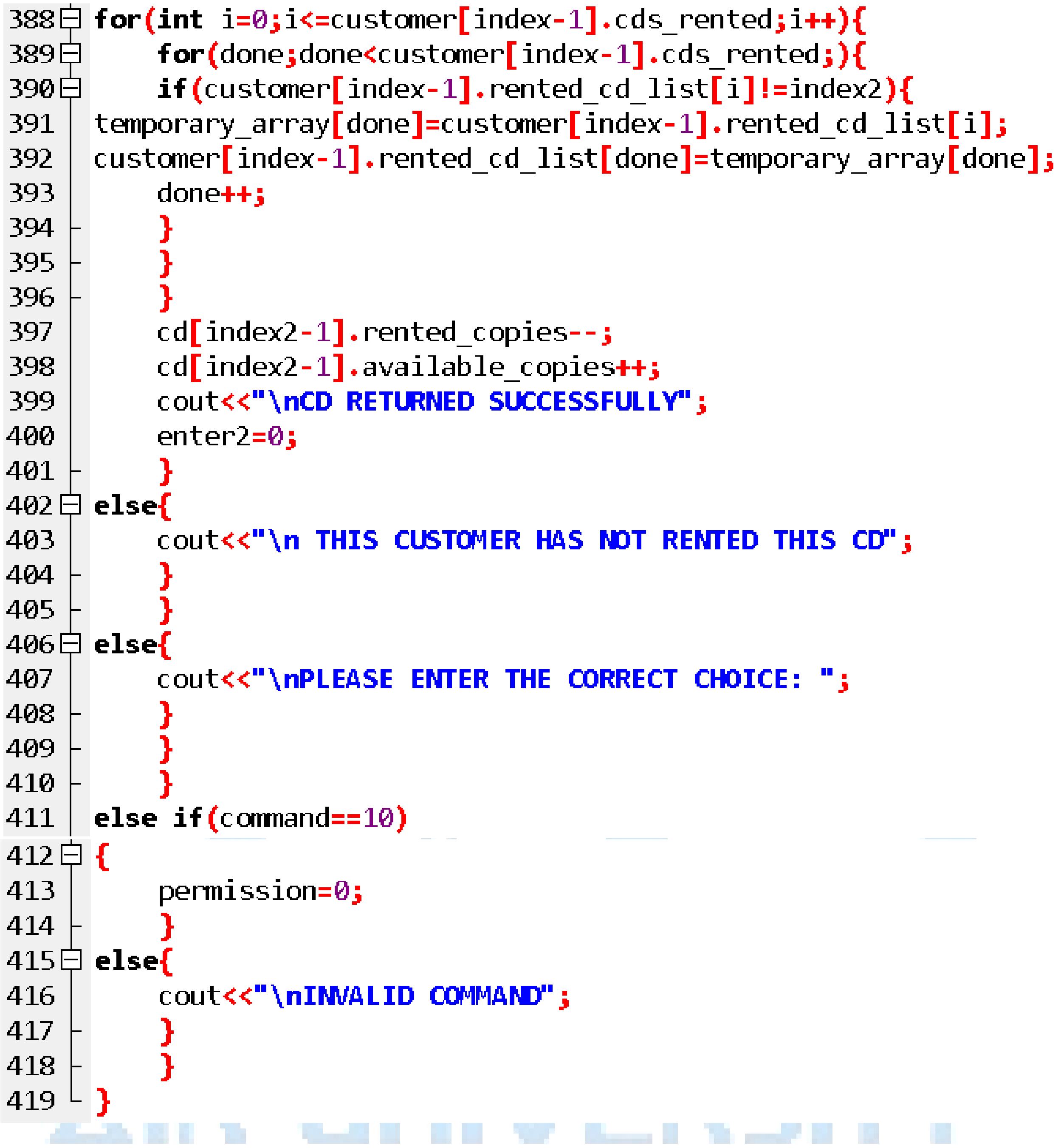
```
	}
	}
	}
	}
	else if(command==8)
	{
	cout<<"\nDISPLAYING CUSTOMERS :";
for(int j=0;j<TOTAL_CUSTOMERS;j++){
	DISPLAY_CUSTOMER2(&customer[j]);
	}
	int enter=1,enter2=1;
	int index;
while(enter==1){
	cout<<"\nENTER THE INDEX NUMBER OF CUSTOMER: ";
	cin>>index;
if(index>0&&index<=TOTAL_CUSTOMERS){
	enter=0;
	}
else{
	cout<<"\n INVALID ENTRY PLEASE RE ENTER";
	}
	}
	cout<<"\t"<<"INDEX"<<"\t"<<"NAME";
for(int i=0;i<TOTAL_CDS;i++){
```



```
	cout<<"\n->\t CD "<<i+1;
	DISPLAY_CD2(&cd[i]);
	}
	while(enter2==1){
	int index2;
	cout<<"\nENTER THE INDEX NUMBER OF CD TO RENT";
	cin>>index2;
if (index2>0&&index2<=TOTAL_CDS){
	if(cd[index2-1].available_copies>0){
	customer[index-1].cds_rented++;
customer[index-1].rented_cd_list
[customer[index-1].cds_rented-1]=index2;
	cd[index2-1].rented_copies++;
	cd[index2-1].available_copies--;
	cout<<"\nCD RENTED SUCCESSFULLY";
	enter2=0;
	}
```

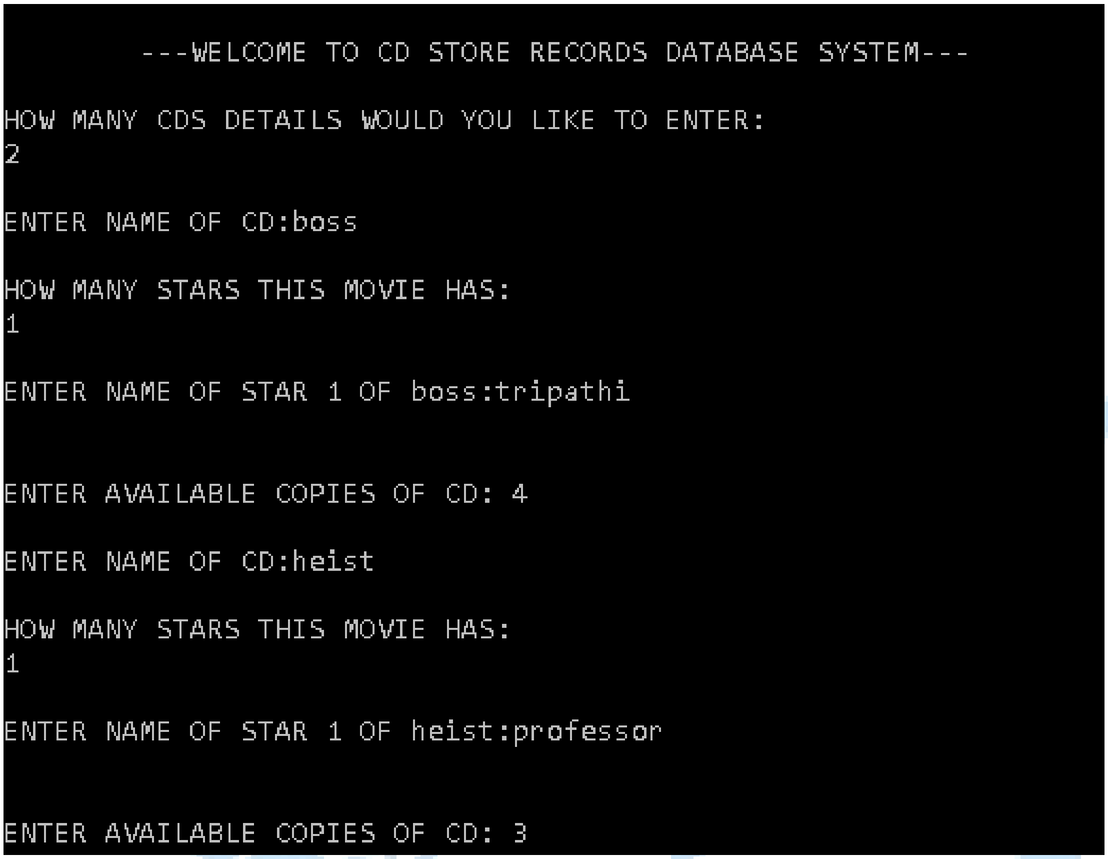


```
else{
    cout<<"\n THE CD YOU WANT TO RENT IS OUT OF STOCK";
    }
    }
else{
    cout<<"\nPLEASE ENTER THE CORRECT CHOICE: ";
    }
    }
    }
else if(command==9)
{
    cout<<"\nDISPLAYING CUSTOMERS :";
for(int j=0;j<TOTAL_CUSTOMERS;j++)
{
    DISPLAY_CUSTOMER2(&customer[j]);
    }
    int enter=1,enter2=1;
    int index;
while(enter==1){
    cout<<"\nENTER THE INDEX NUMBER OF CUSTOMER: ";
    cin>>index;
if(index>0&&index<=TOTAL_CUSTOMERS){
    enter=0;
    }
else{
    cout<<"\n INVALID ENTRY PLEASE RE ENTER";
    }
    }
for(int i=0;i<customer[index-1].cds_rented;i++){
    cout<<"\nRENTED CD "<<i+1<<" is: "
    <<cd[customer[index-1].rented_cd_list[i]-1].name;
    }
while(enter2==1){
    int index2;
    cout<<"\nENTER THE INDEX NUMBER OF CD TO RETURN";
    cin>>index2;
if (index2>0&&index2<=TOTAL_CDS){
    if(cd[index2-1].rented_copies>0){
    customer[index-1].cds_rented--;
    int temporary_array[customer[index-1].cds_rented];
    int done=0;
```



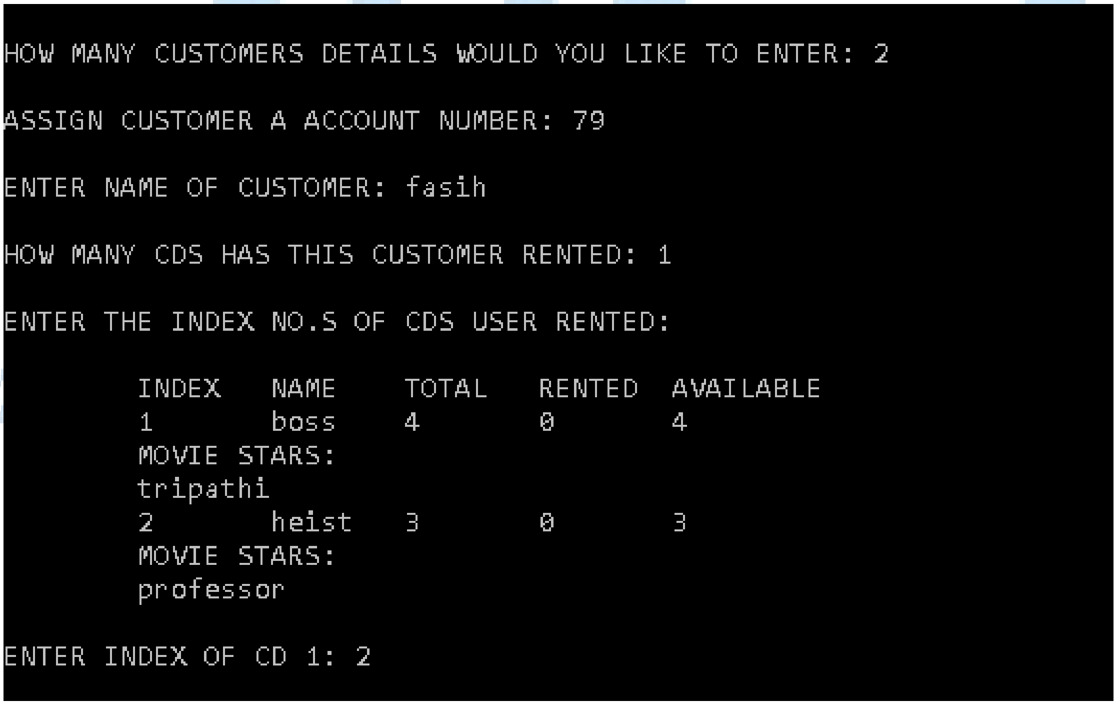
```
for(int i=0;i<=customer[index-1].cds_rented;i++){
    for(done;done<customer[index-1].cds_rented;){
    if(customer[index-1].rented_cd_list[i]!=index2){
temporary_array[done]=customer[index-1].rented_cd_list[i];
customer[index-1].rented_cd_list[done]=temporary_array[done];
    done++;
    }
    }
    }
    cd[index2-1].rented_copies--;
    cd[index2-1].available_copies++;
    cout<<"\nCD RETURNED SUCCESSFULLY";
    enter2=0;
    }
else{
    cout<<"\n THIS CUSTOMER HAS NOT RENTED THIS CD";
    }
    }
else{
    cout<<"\nPLEASE ENTER THE CORRECT CHOICE: ";
    }
    }
    }
else if(command==10)
{
    permission=0;
    }
else{
    cout<<"\nINVALID COMMAND";
    }
    }
}
```

OUTPUT



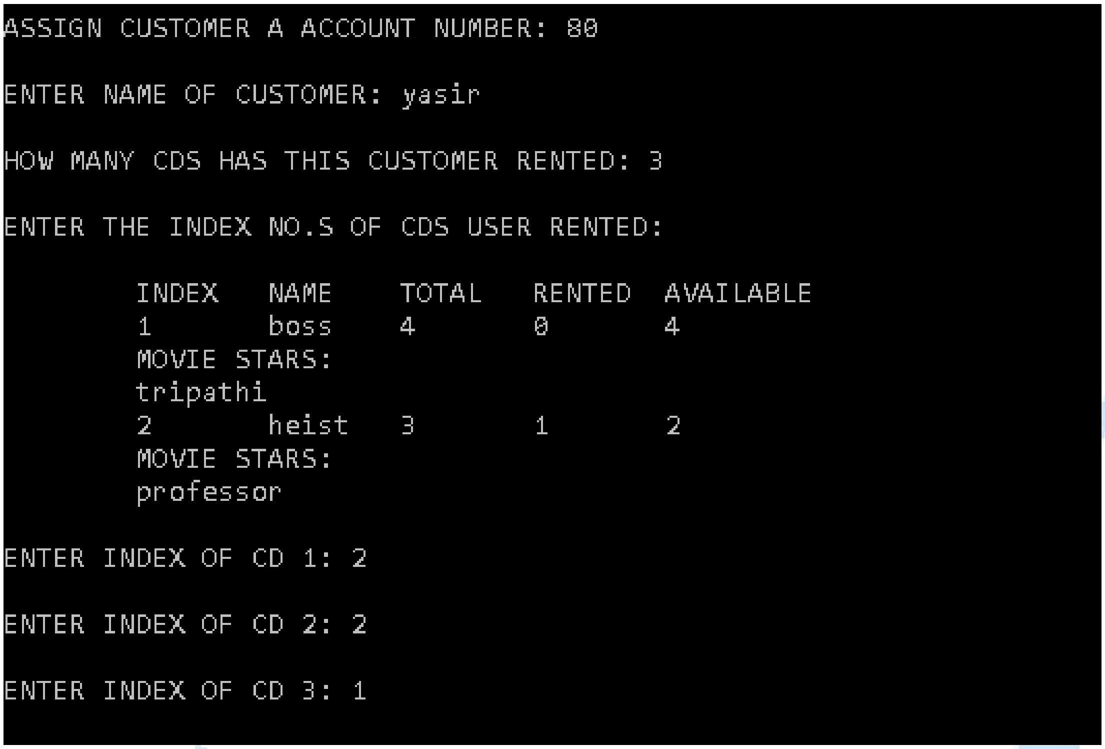
```
        ---WELCOME TO CD STORE RECORDS DATABASE SYSTEM---

HOW MANY CDS DETAILS WOULD YOU LIKE TO ENTER:
2

ENTER NAME OF CD:boss

HOW MANY STARS THIS MOVIE HAS:
1

ENTER NAME OF STAR 1 OF boss:tripathi


ENTER AVAILABLE COPIES OF CD: 4

ENTER NAME OF CD:heist

HOW MANY STARS THIS MOVIE HAS:
1

ENTER NAME OF STAR 1 OF heist:professor


ENTER AVAILABLE COPIES OF CD: 3
```

CD Details Input



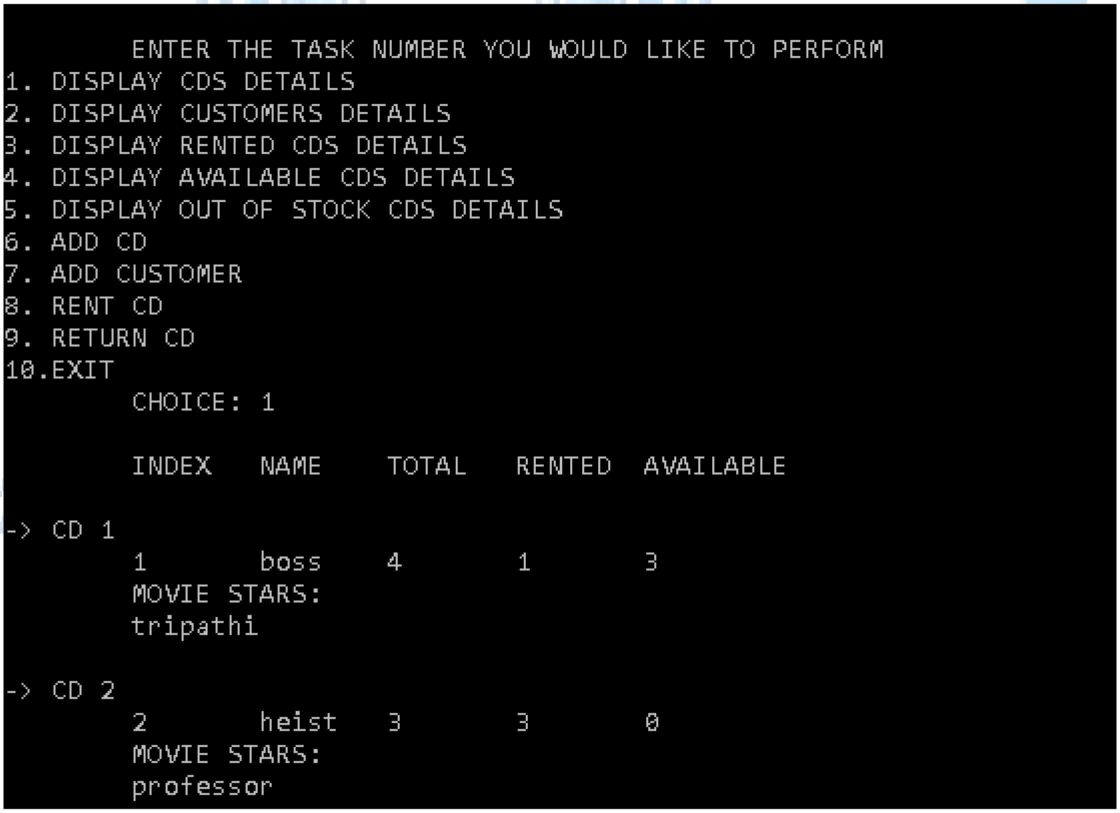
```
HOW MANY CUSTOMERS DETAILS WOULD YOU LIKE TO ENTER: 2

ASSIGN CUSTOMER A ACCOUNT NUMBER: 79

ENTER NAME OF CUSTOMER: fasih

HOW MANY CDS HAS THIS CUSTOMER RENTED: 1

ENTER THE INDEX NO.S OF CDS USER RENTED:

        INDEX   NAME    TOTAL   RENTED  AVAILABLE
        1       boss    4       0       4
        MOVIE STARS:
        tripathi
        2       heist   3       0       3
        MOVIE STARS:
        professor

ENTER INDEX OF CD 1: 2
```

Detail of First costumer and its rented Cds



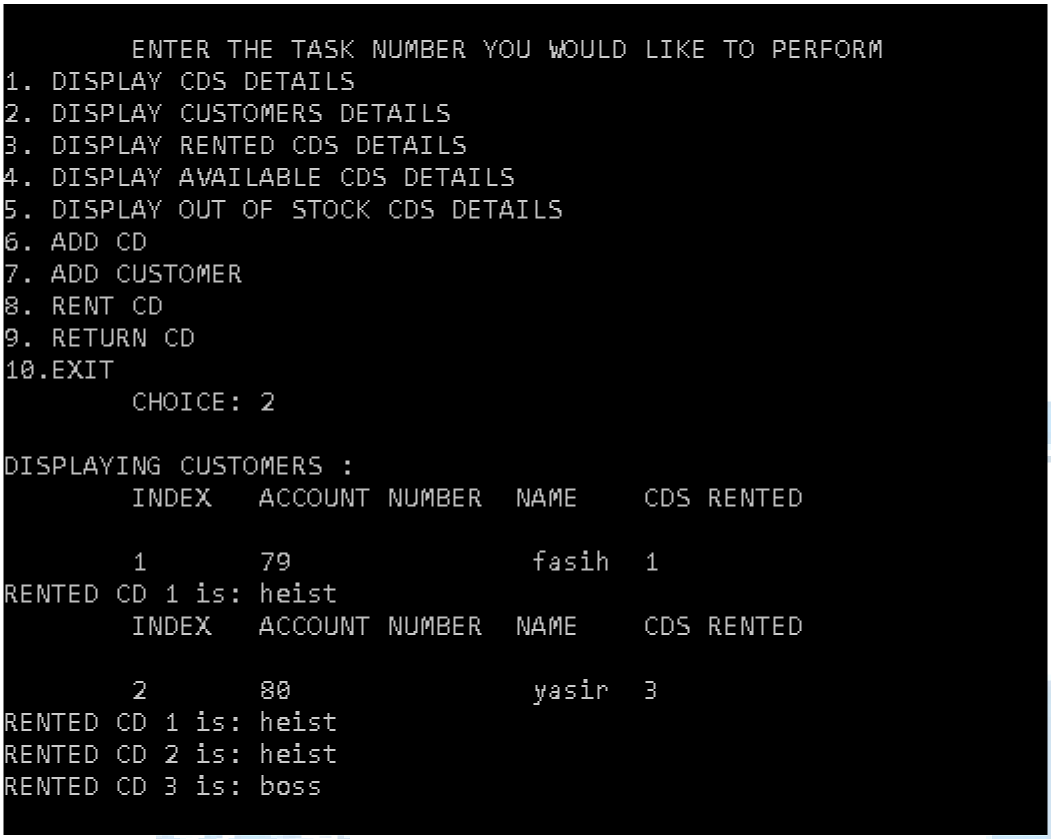
```
ASSIGN CUSTOMER A ACCOUNT NUMBER: 80

ENTER NAME OF CUSTOMER: yasir

HOW MANY CDS HAS THIS CUSTOMER RENTED: 3

ENTER THE INDEX NO.S OF CDS USER RENTED:

        INDEX   NAME    TOTAL   RENTED  AVAILABLE
        1       boss    4       0       4
        MOVIE STARS:
        tripathi
        2       heist   3       1       2
        MOVIE STARS:
        professor

ENTER INDEX OF CD 1: 2

ENTER INDEX OF CD 2: 2

ENTER INDEX OF CD 3: 1
```

Details of Second costumer and its rented CDs



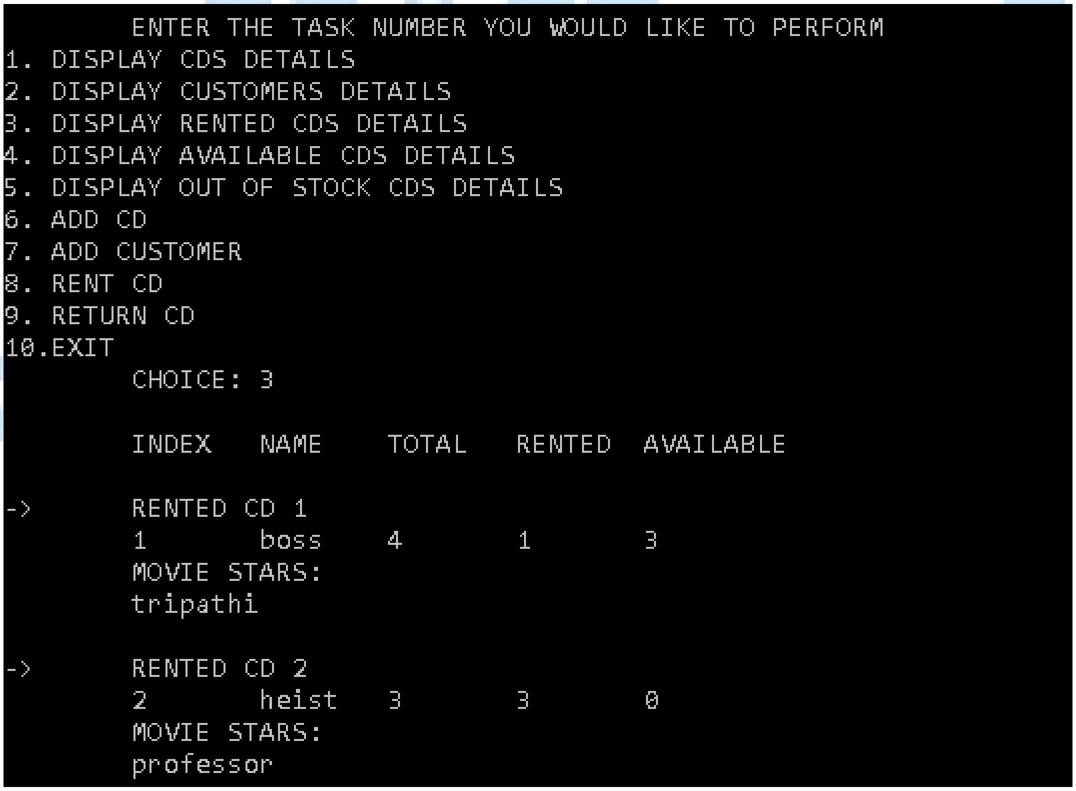
```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 1

        INDEX   NAME    TOTAL   RENTED  AVAILABLE

-> CD 1
        1       boss    4       1       3
        MOVIE STARS:
        tripathi

-> CD 2
        2       heist   3       3       0
        MOVIE STARS:
        professor
```

Display Cd Details



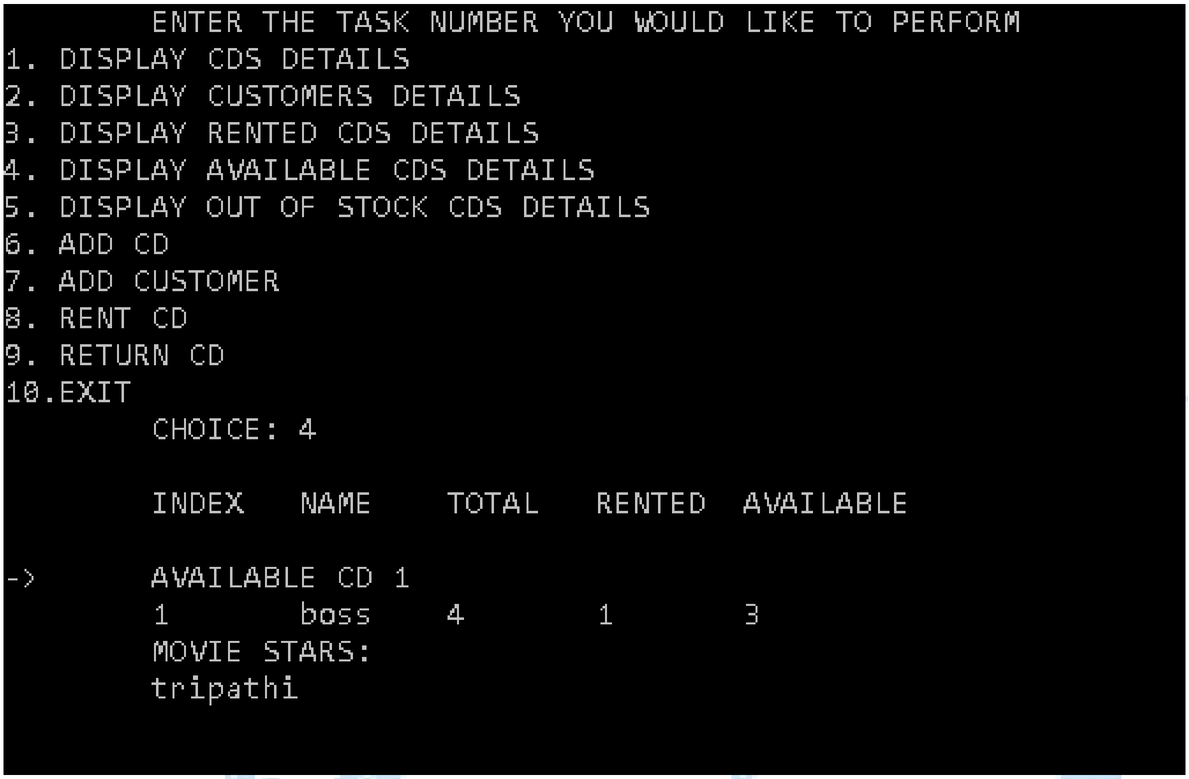
```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 2

DISPLAYING CUSTOMERS :
        INDEX   ACCOUNT NUMBER  NAME    CDS RENTED

        1       79               fasih  1
RENTED CD 1 is: heist
        INDEX   ACCOUNT NUMBER  NAME    CDS RENTED

        2       80               yasir  3
RENTED CD 1 is: heist
RENTED CD 2 is: heist
RENTED CD 3 is: boss
```

Display Costumer Details



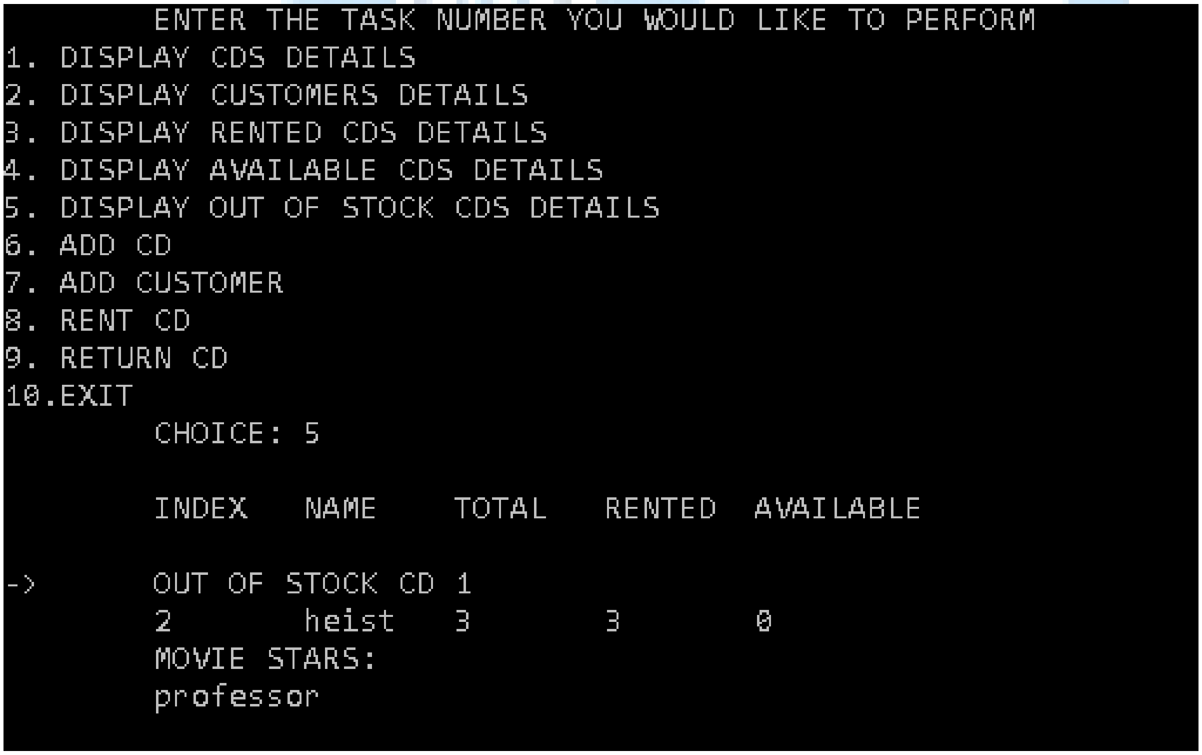
```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 3

        INDEX   NAME    TOTAL   RENTED  AVAILABLE

->      RENTED CD 1
        1       boss    4       1       3
        MOVIE STARS:
        tripathi

->      RENTED CD 2
        2       heist   3       3       0
        MOVIE STARS:
        professor
```

Display rented cd list



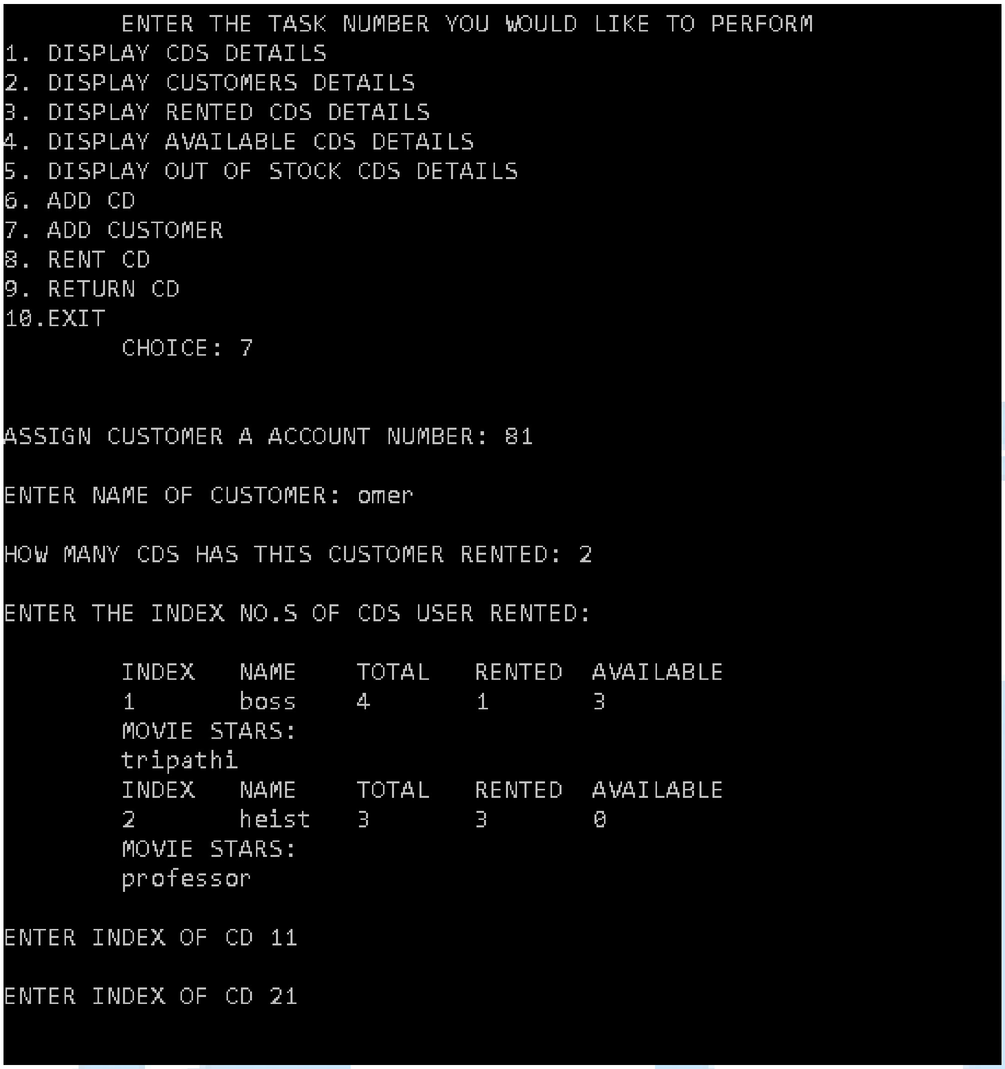
```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 4

        INDEX   NAME    TOTAL   RENTED  AVAILABLE

->      AVAILABLE CD 1
        1       boss    4       1       3
        MOVIE STARS:
        tripathi
```

Minor Details of Cd



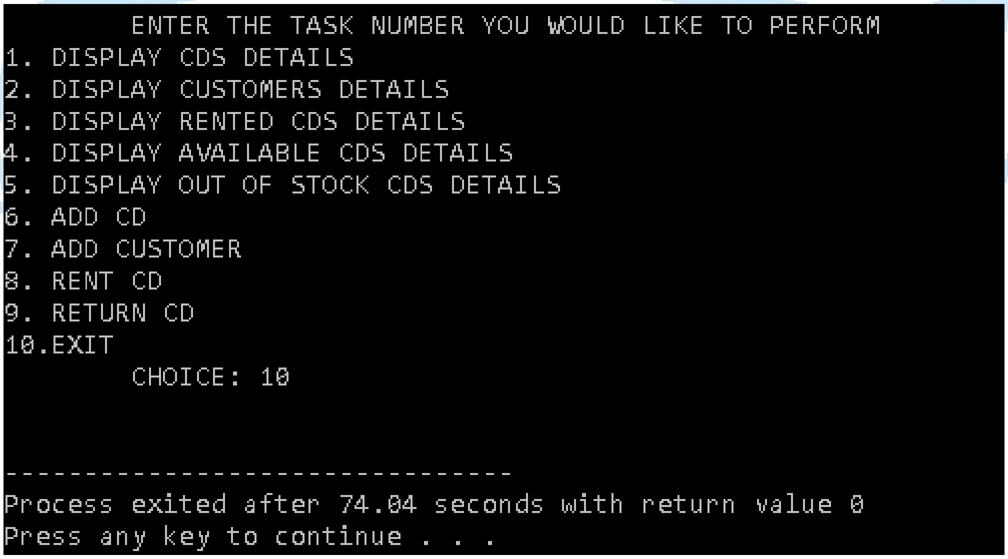
```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 5

        INDEX   NAME    TOTAL   RENTED  AVAILABLE

->      OUT OF STOCK CD 1
        2       heist   3       3       0
        MOVIE STARS:
        professor
```

Display out of stock CD details



```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 7


ASSIGN CUSTOMER A ACCOUNT NUMBER: 81

ENTER NAME OF CUSTOMER: omer

HOW MANY CDS HAS THIS CUSTOMER RENTED: 2

ENTER THE INDEX NO.S OF CDS USER RENTED:

        INDEX   NAME    TOTAL   RENTED  AVAILABLE
        1       boss    4       1       3
        MOVIE STARS:
        tripathi
        INDEX   NAME    TOTAL   RENTED  AVAILABLE
        2       heist   3       3       0
        MOVIE STARS:
        professor

ENTER INDEX OF CD 11

ENTER INDEX OF CD 21
```

Add Costumer Details



```
        ENTER THE TASK NUMBER YOU WOULD LIKE TO PERFORM
1. DISPLAY CDS DETAILS
2. DISPLAY CUSTOMERS DETAILS
3. DISPLAY RENTED CDS DETAILS
4. DISPLAY AVAILABLE CDS DETAILS
5. DISPLAY OUT OF STOCK CDS DETAILS
6. ADD CD
7. ADD CUSTOMER
8. RENT CD
9. RETURN CD
10.EXIT
        CHOICE: 10



--------------------------------
Process exited after 74.04 seconds with return value 0
Press any key to continue . . .
```

Exit The Program

CONCLUSION:

- In this project we had learned how to make a CD Management System. This Software helps the shop owner to save his time in doing check and balance on the shop record.
- This is to conclude that the project that we undertook was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirements which have been remaining, can be completed with a short extension.
- The further implementation is performing in this software are connectivity to internet that the customer detail so online and every detail related with organization are also so on the internet.
- The main technical field of this project that can be developed after the enhanced version of this software is that it can calculate the profit or loss of the organization that help to get the financial position of the organization.