1. **indicate which options are completed**

Everything is completed

1. **indicate which options are not completed**

None

1. **indicate which parts work as expected/gives correct output**

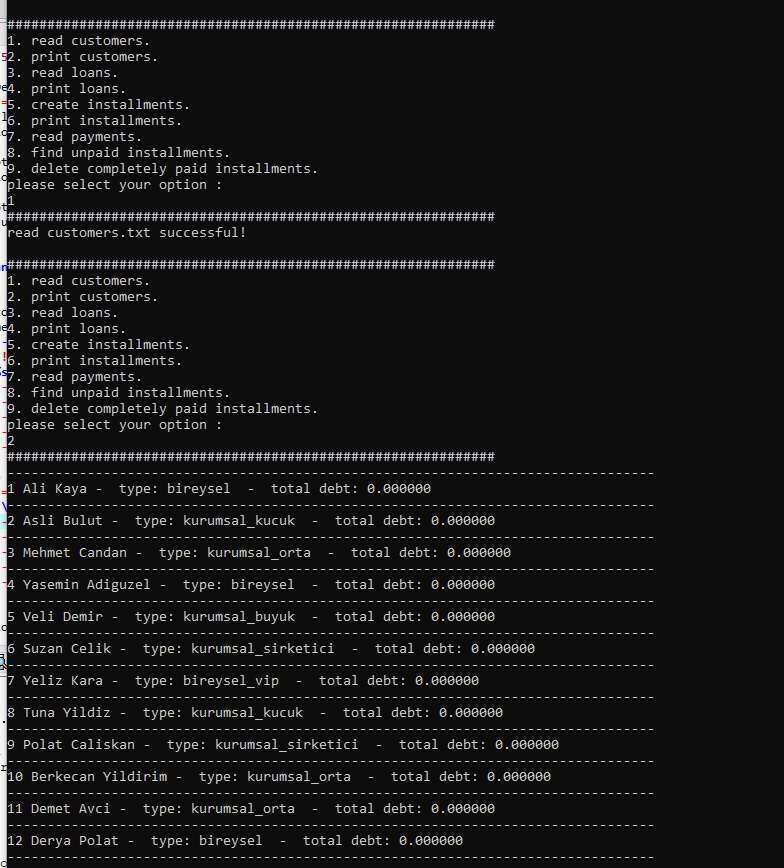
Everything work correctly

1. **indicate which options do not work/unexpected output**

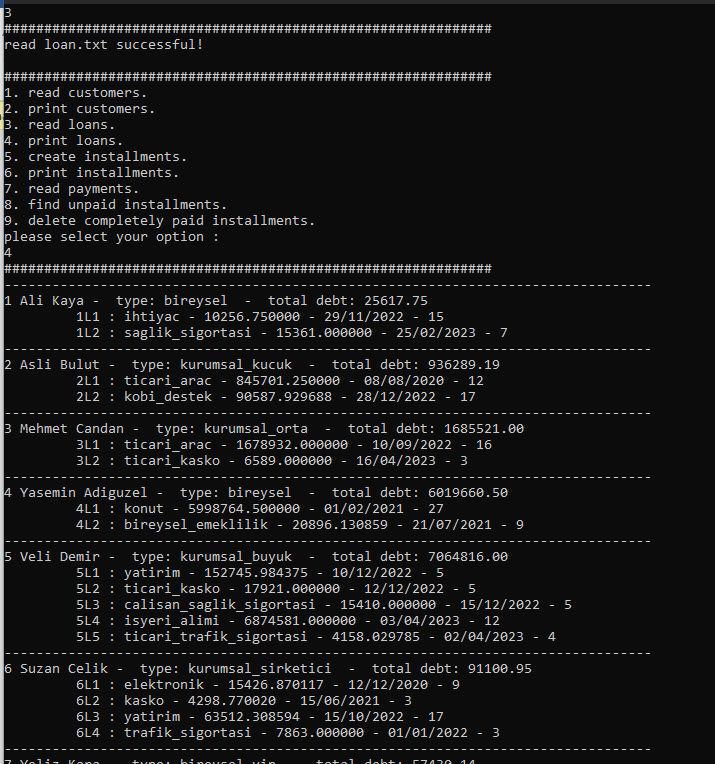
None

1. **Provide screenshot of output for each option**

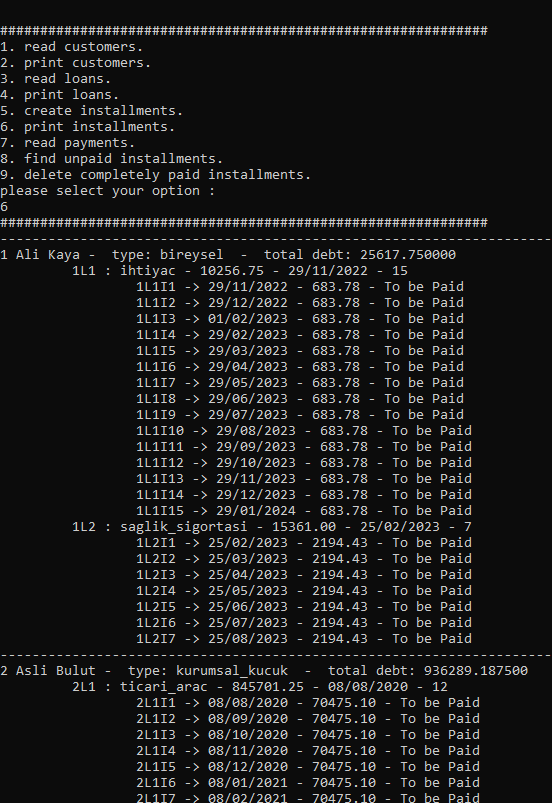
**1 – 2**



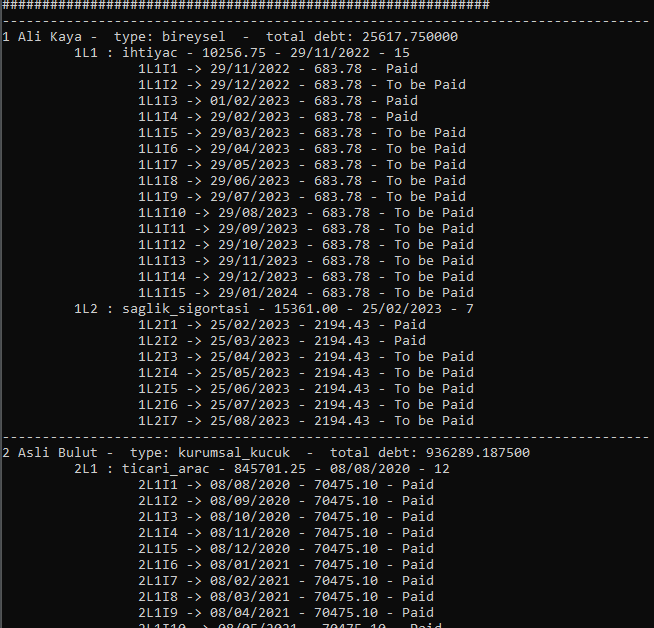
3-4



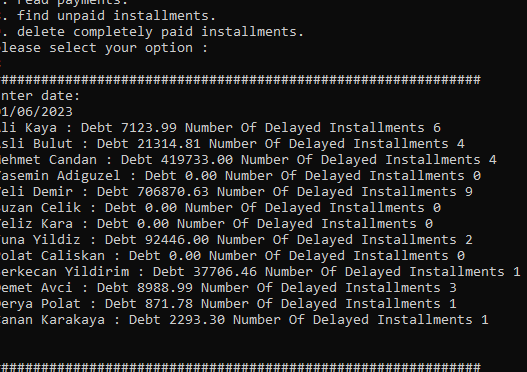
5-6



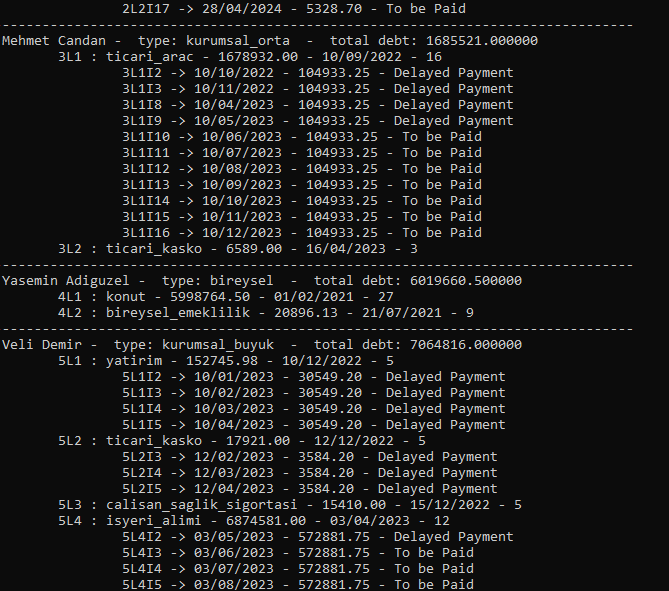
7 – 6



8



9 – 6



1. **Write explanations about the functions you have written. Include the purpose of the function and how it does which task.**

void customer\_append(Customer\*\* head\_ref, char\* name,char\* surname,int customerid,char\* customertype, double totaldebt);

add a customer node to the end of customer linked list

void read\_customers(Customer\*\* head);

read customers data from files ,then add all of customers to linked list

void printCustomers(Customer\* head);

print customers

int customersSize(Customer\* head);

find total customers

Customer\* getCustomer(Customer\* head,char\* name,char\* surname);

Find customer by name and surname

void printLoans(Customer\* head);

print loans

Loan\* loan\_append(Loan\* head\_ref, char\* loanid,char\* type,float totalamount,int totalinstallmentnum, char\* processdate);

add a loan node to the loan linked list, (sorted insert )

Installment\* installment\_append(Installment\* head\_ref, char\* insid, char\* installmentdate,float amount,short ispaid );

add a installment node to the end of installment linked list

int compareDate(char\* d1,char\* d2);

compare 2 date string

void read\_loans(Customer\*\* head);

read data from loan.txt

void createInstallments(Customer\* head);

create installments for each loan

char\* addMonth(char\* date,int monthToAdd);  
 use for : add 1 moth to date

void printInstallments(Customer\* head);

print installmments

void read\_payment(Customer\*\* head);

read payments.txt

Loan\* getLoan(Customer\* head,char\* id);

Find loan by loanID

void payALl(Loan\* loan);

pay all installments

void payAt(Loan\* loan,int at);

pay installments by number

void findUnpaidInstallments(Customer\* head);

find unpaid installments

void payDelay(Customer\* head,char\* date);

use for findUnpaidInstallments

change isPaid = 3 for delay installments

Installment\* findPaidIns(Installment\* head\_ref);

Find paid installment

Installment\* DeleteInstallment(Installment\* head\_ref,Installment\* deleteNode);

Delete a node in installments linked list

void deletePaidInstallments(Customer\* head);

delete all paid installments

1. **Explain where and how the linked lists are created and elements of a linked list are deleted. If a linked list has a sorted structure, please explain where and how it is sorted.**

**linked list are deleted:**

+ in delete paid installment

Find exist paid installment node

Delete that node

**lists are created:**

+ in read customers,loans,payments

+ each line add a new node to the linked list

**list sorted:**

+ in insert loan

+ compare the processdate to each node in the linked list then place it at correct order!