BABATI URBAN COUNCIL
REVENUE COLLECTION SYSTEM

LEE BIRIR
COMPUTER STUDIES

2006

A dissertation submitted in partial fulfillment of the requirements for the KCSE examination in Computer Studies

ACKNOWLEDGEMENTS

I would like to thank my parents, Mrs. Priscilla Birir & Mr. Tim Birir for offering me moral support throughout the project. I also thank my Computer Studies teacher, Mr. Kyalo for advice and guidance throughout the course. Last but not least, I thank God for his wisdom and keeping me well throughout the development of the program.

TABLE OF CONTENTS

1. INTRODUCTION

Babati Urban Council Revenue Collection System is to computerize its various revenue collection operations. This program has been written with the aim of automating the council's revenue collecting system. The program will carry out collecting of revenue of the various services the council offers its residents automatically. The services include: housing, garbage collection and disposal, sewerage services, licensing of businesses and maintenance of council property such as market charges and parking fees.

The Analysis contains a detailed description of the current system, the broad aims of the program and the advantages and disadvantages of the new and old systems.

The Design section has the system flowchart and all the module flowcharts, which represent the basic working of the program. It also has details on the approach that was used during the programming.

The Coding chapter contains all the instructions that were written in the program itself.

Testing and Demonstration has explanations on the working of the program that are illustrated by screen captures.

The User Manual shows the person operating the program how to use it.

2. PROBLEM DEFINITION

Babati Urban Council Revenue Collection System currently carries out all its operations manually. The operations include:

- Registering of residents in the council
- Collecting of revenue of the various services offered by the Council
- Maintenance of lists showing information of residents and businesses registered
- Determining the amount of revenue collected in a day, month or year
- Offering services such as:
    - Housing
    - Garbage collection and disposal
    - Sewerage services
    - Licensing of businesses
    - Maintenance of Council property

When a tenant is registered he/she inputs his/her ID number, which identifies him/her in all the transactions he/she is involved in.
Revenue is collected daily, monthly or annually according to the type of service against the resident's ID number.
There are lists to show the residents' and businesses' details and also the revenue collected. Records of details and revenue collection should be kept.
The revenue of the various services is collected at fixed terms according to the rates established by the Council.
Total amount of revenue collected in a day, month or year can be determined automatically.

3. ANALYSIS

Babati Urban Council revenue collection system isn't computerized and all its operations are done manually by people. Information concerning revenue collection is stored in physical files and all revenue is calculated manually by manual labor. The advantage of this system is that it doesn't need electricity. However it has quite a number of disadvantages. The employees of the revenue collection system usually take a long time to do the revenue calculations and at times the calculations have errors. The residents' records also take a lot of space thus force the Council to build more stores and this is not cost effective. The residents' records also don't have backups thus are prone to great threat if there is data loss. In the long run, the current manual system is inefficient.

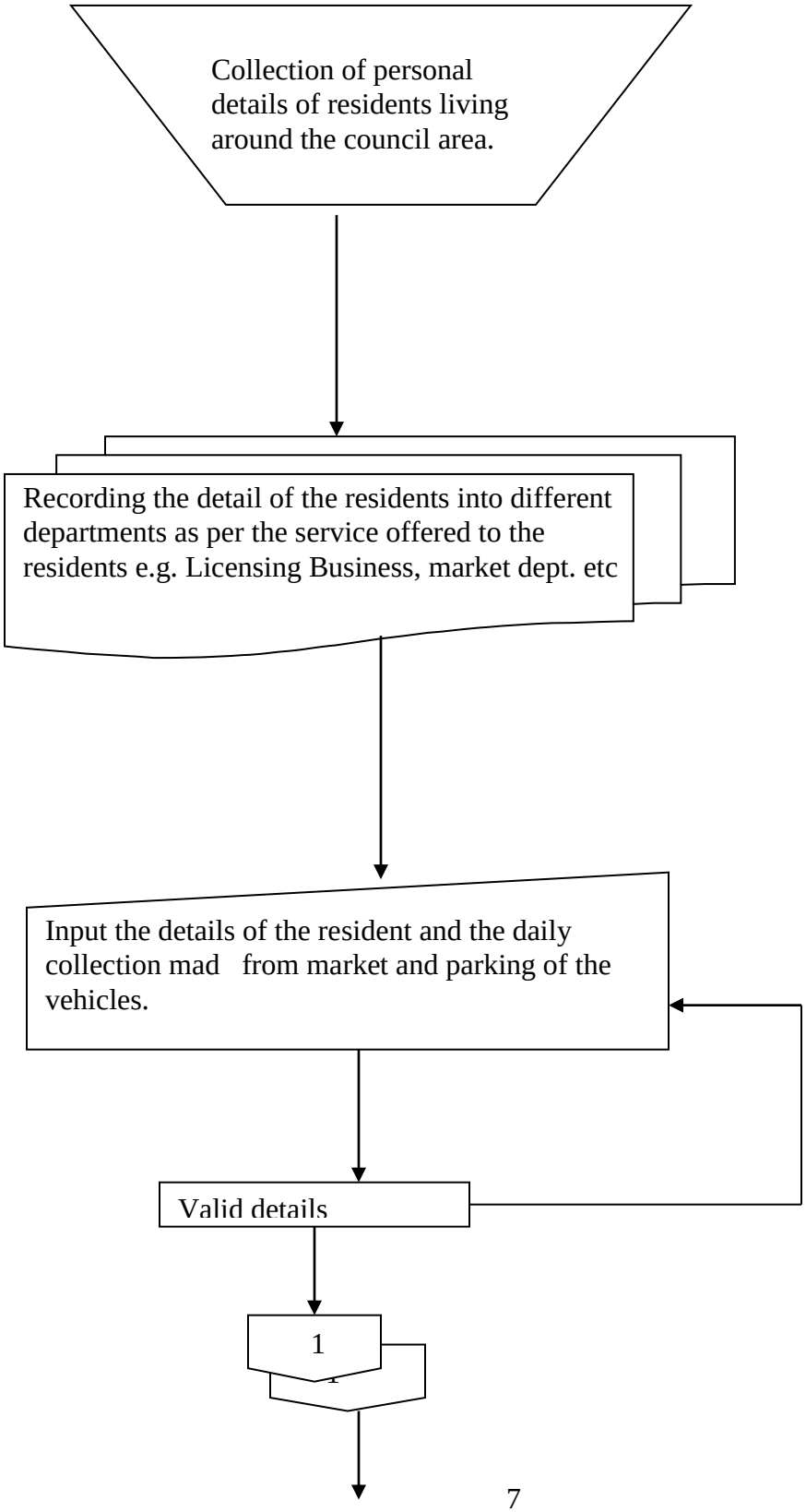
Once computerized, Babati Urban Council revenue collection system will be able to:

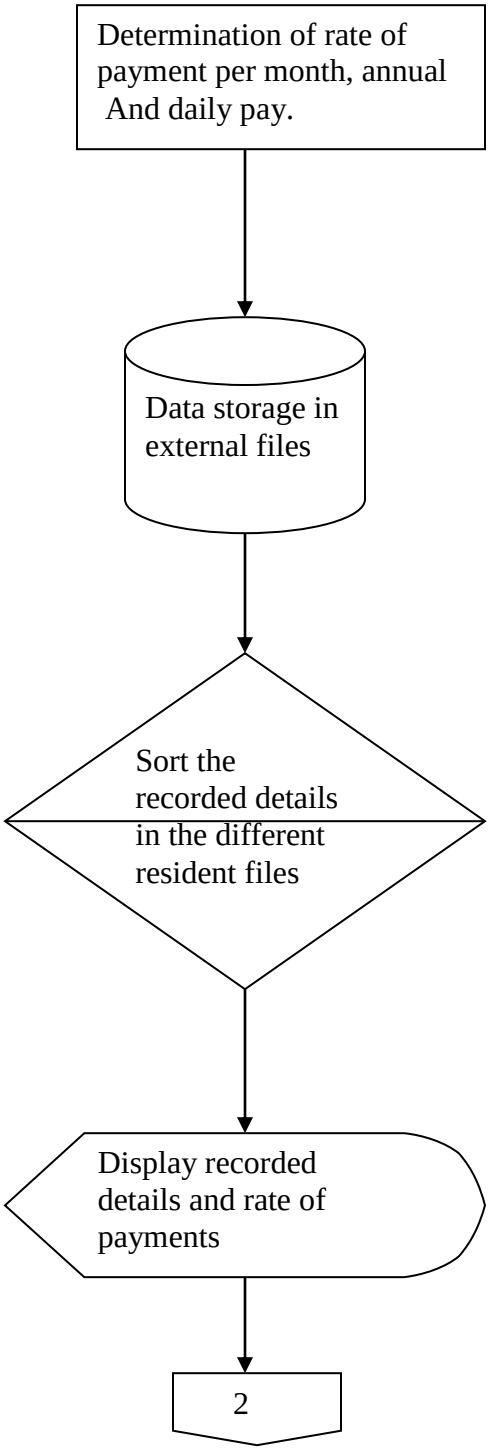
- Store information such as resident and business records in computer files
- Perform calculations on the residents' and businesses' payments and council property automatically

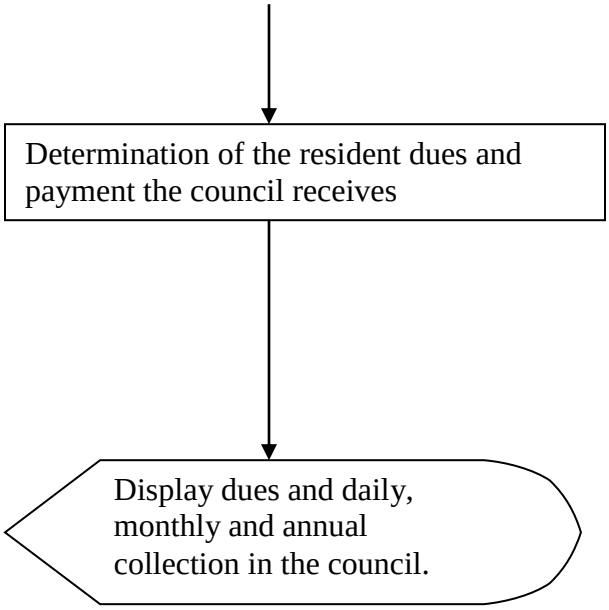
This will be advantageous since the energy and time used to carry out these tasks will be saved. A lot of space will be saved as the records won't be stored in physical files but on computer storage devices such as diskettes and hard disks.
All the calculations done on the program will be accurate and fast since the formulae are set in the program's instructions.
The computerized system will be more cost effective as the manpower needed to carry out the operations won't be needed anymore as one computer can replace several people to do the same job.
The disadvantage of this system is that it depends on electricity, which won't be efficient when there is a power loss. The computers might also be infected with viruses, which can corrupt the records. The computer will only be efficient if the viruses are repaired and when there is regular servicing of the computers.

4. DESIGN

4.1 System Flowchart

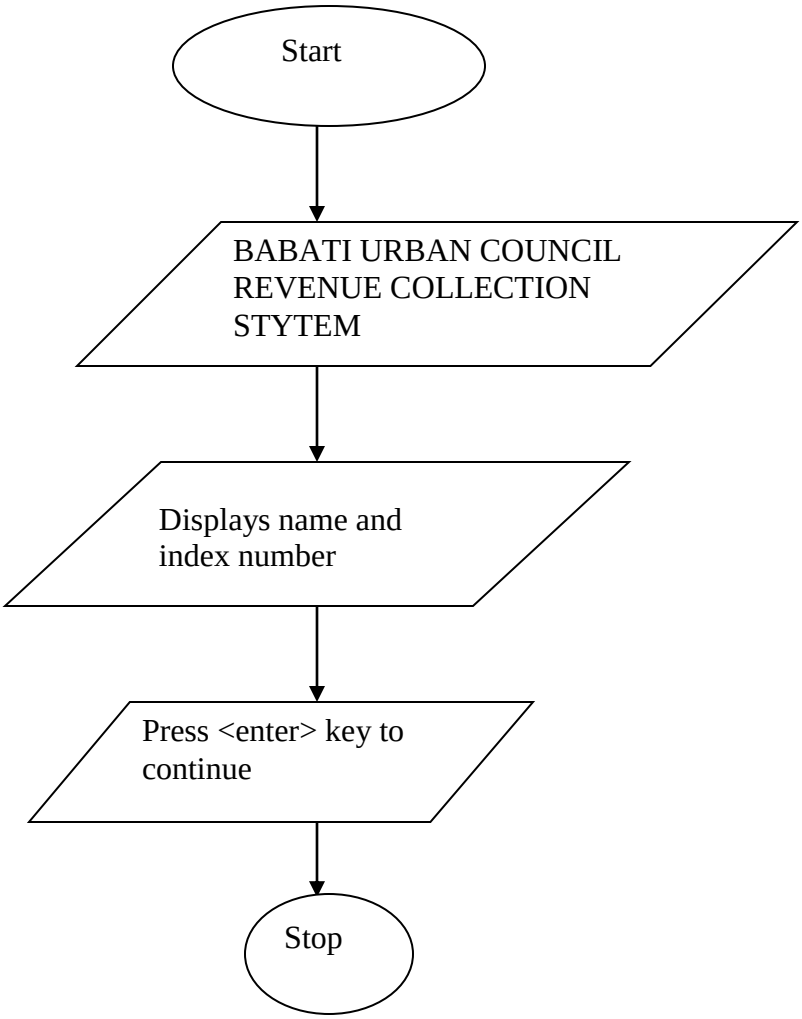



Determination of rate of payment per month, annual And daily pay.
Data storage in external files
Sort the recorded details in the different resident files
Display recorded details and rate of payments
2
2

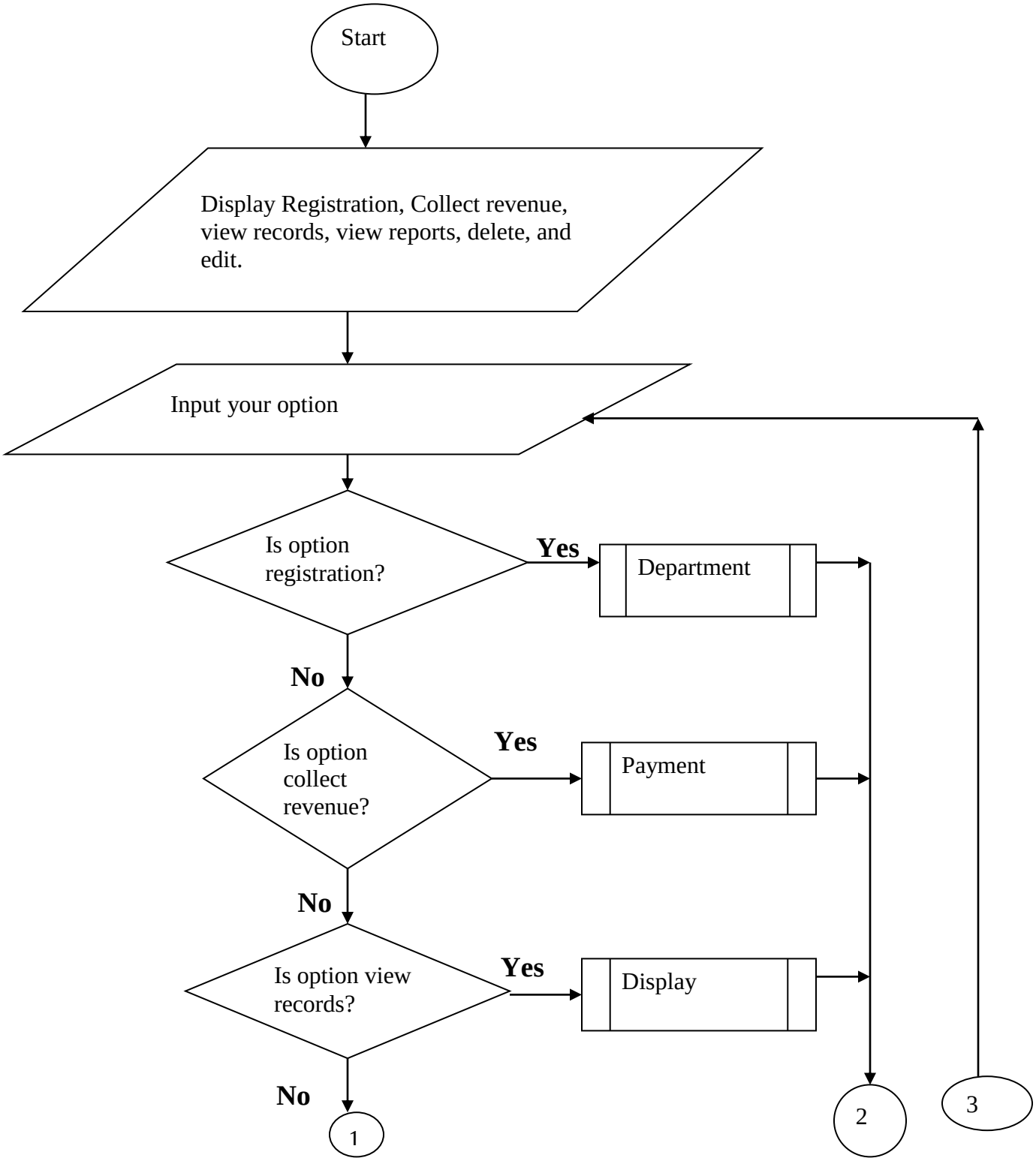

Determination of the resident dues and payment the council receives
Display dues and daily, monthly and annual collection in the council.

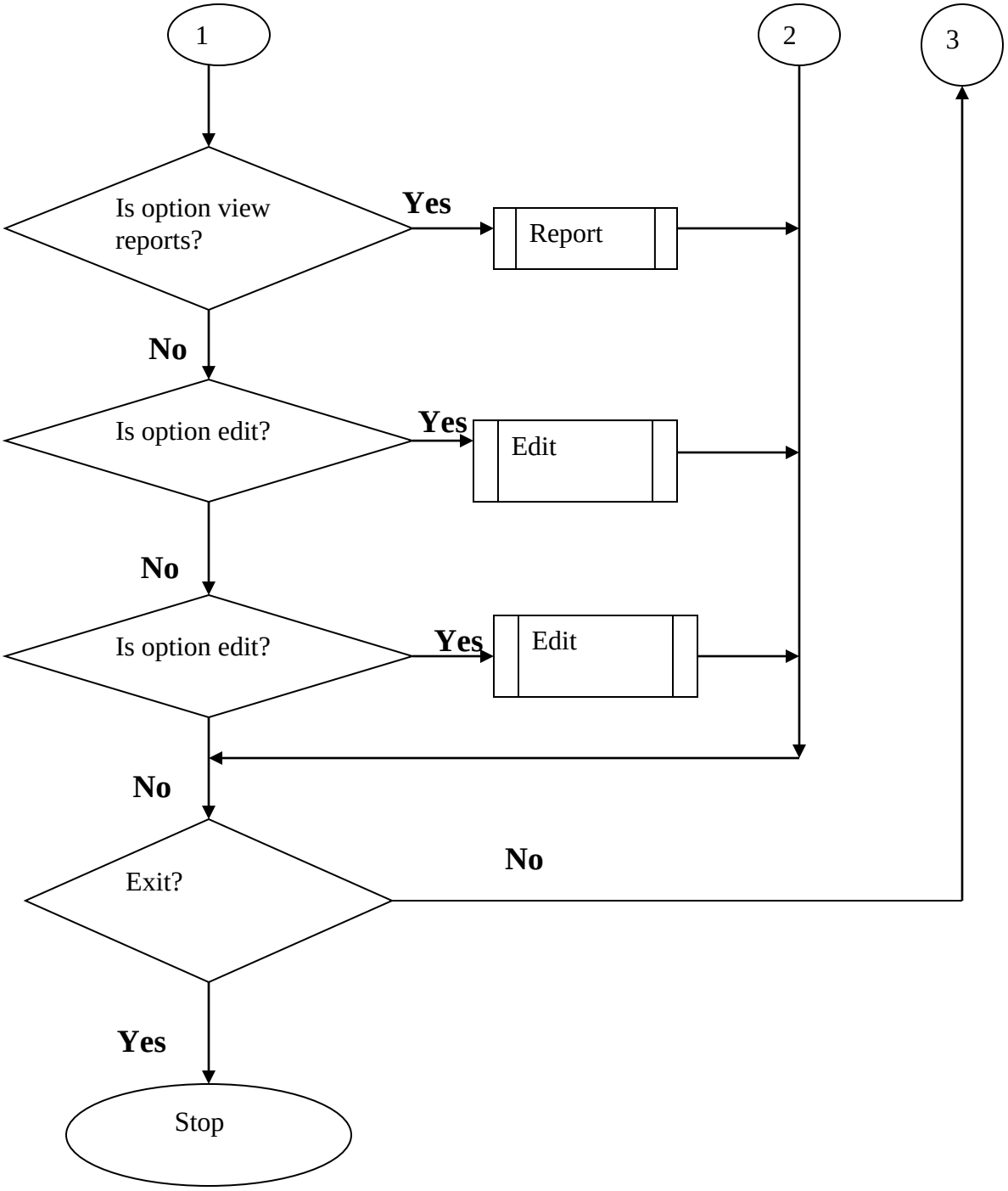
4.2. Module Flowcharts

4.2.1 Procedure Entry

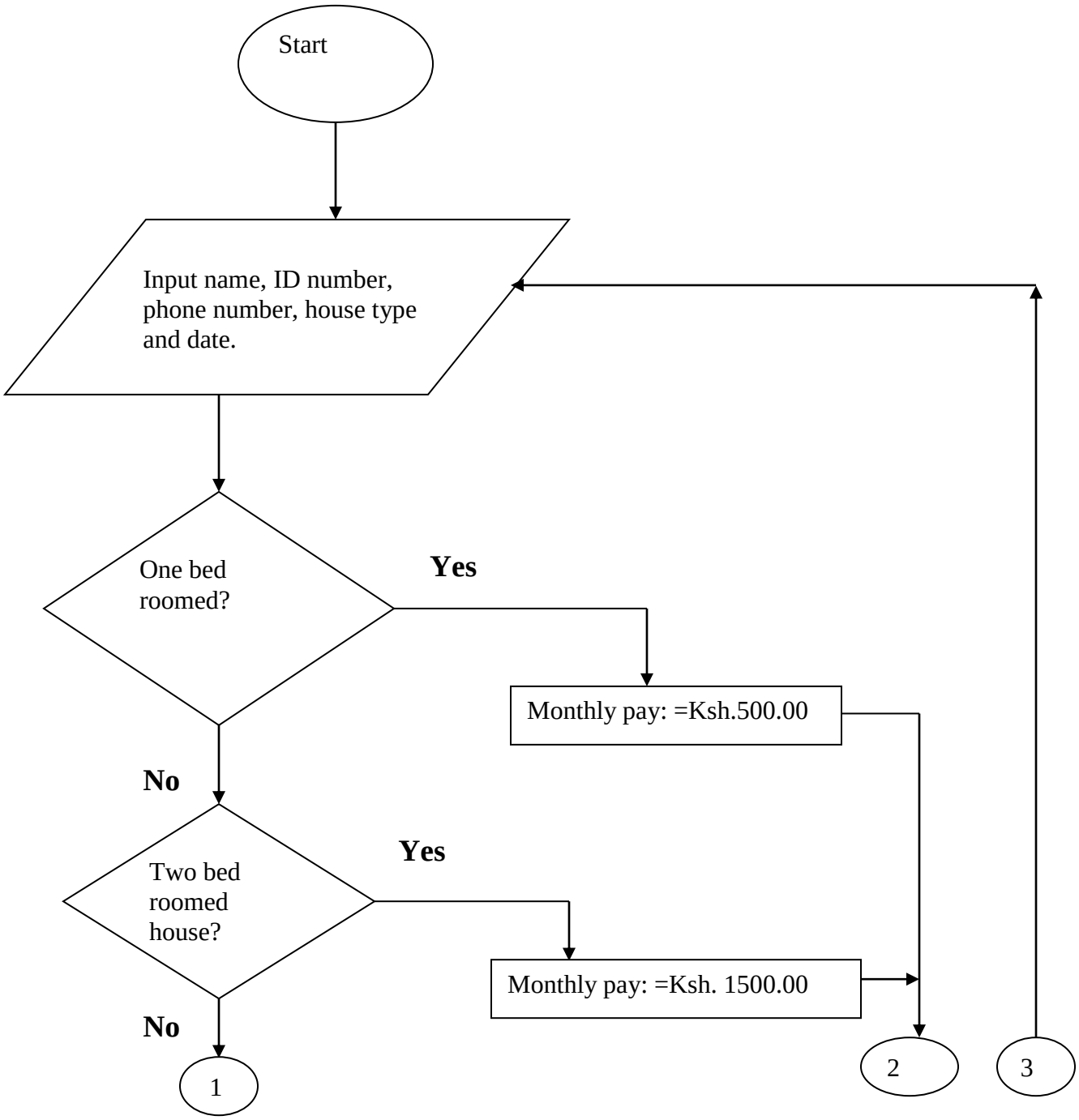


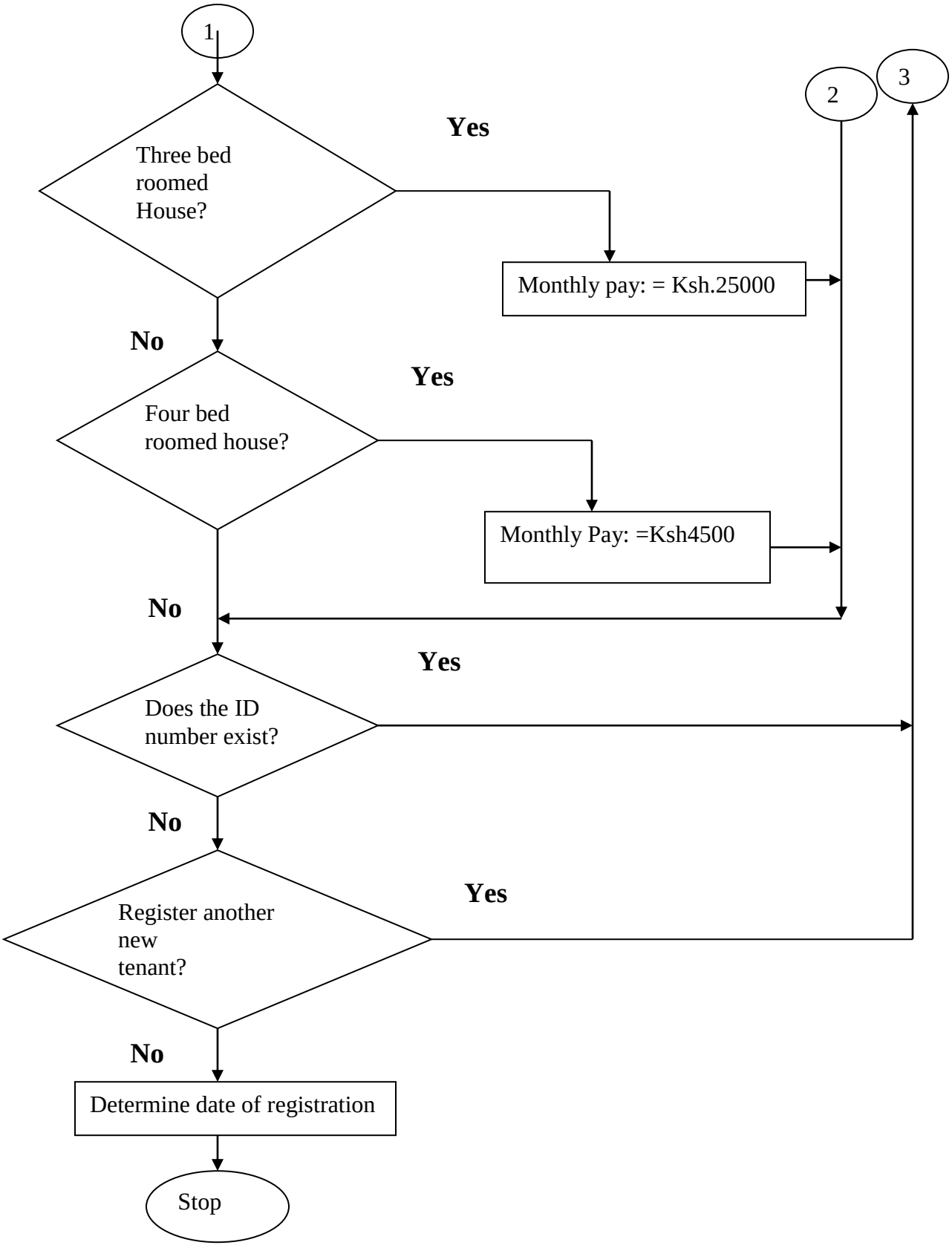
4.2.2 Procedure main menu



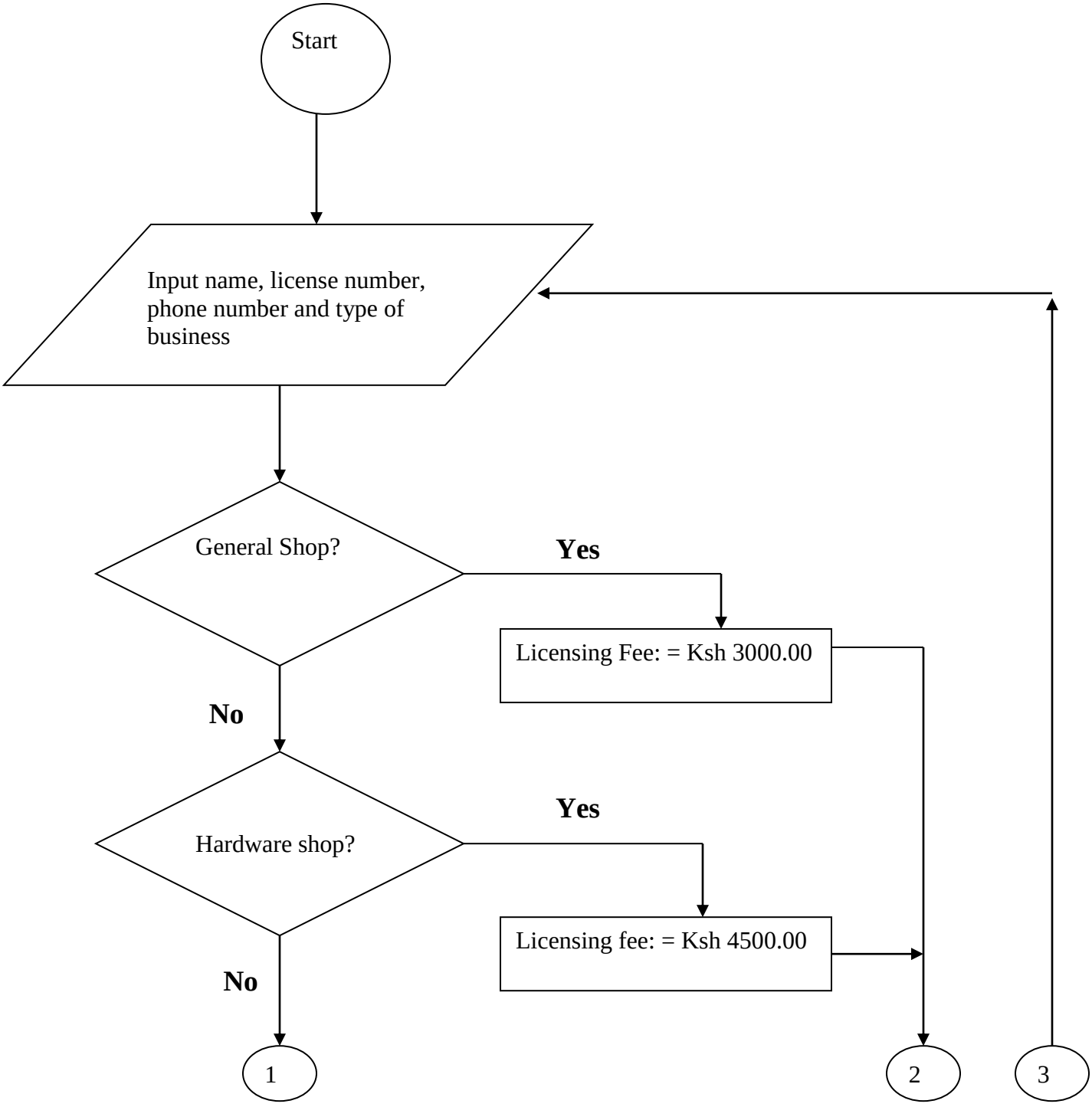

1
2
3
Is option view reports?
Yes
Report
No
Is option edit?
Yes
Edit
No
Is option edit?
Yes
Edit
No
Exit?
No
Yes
Stop

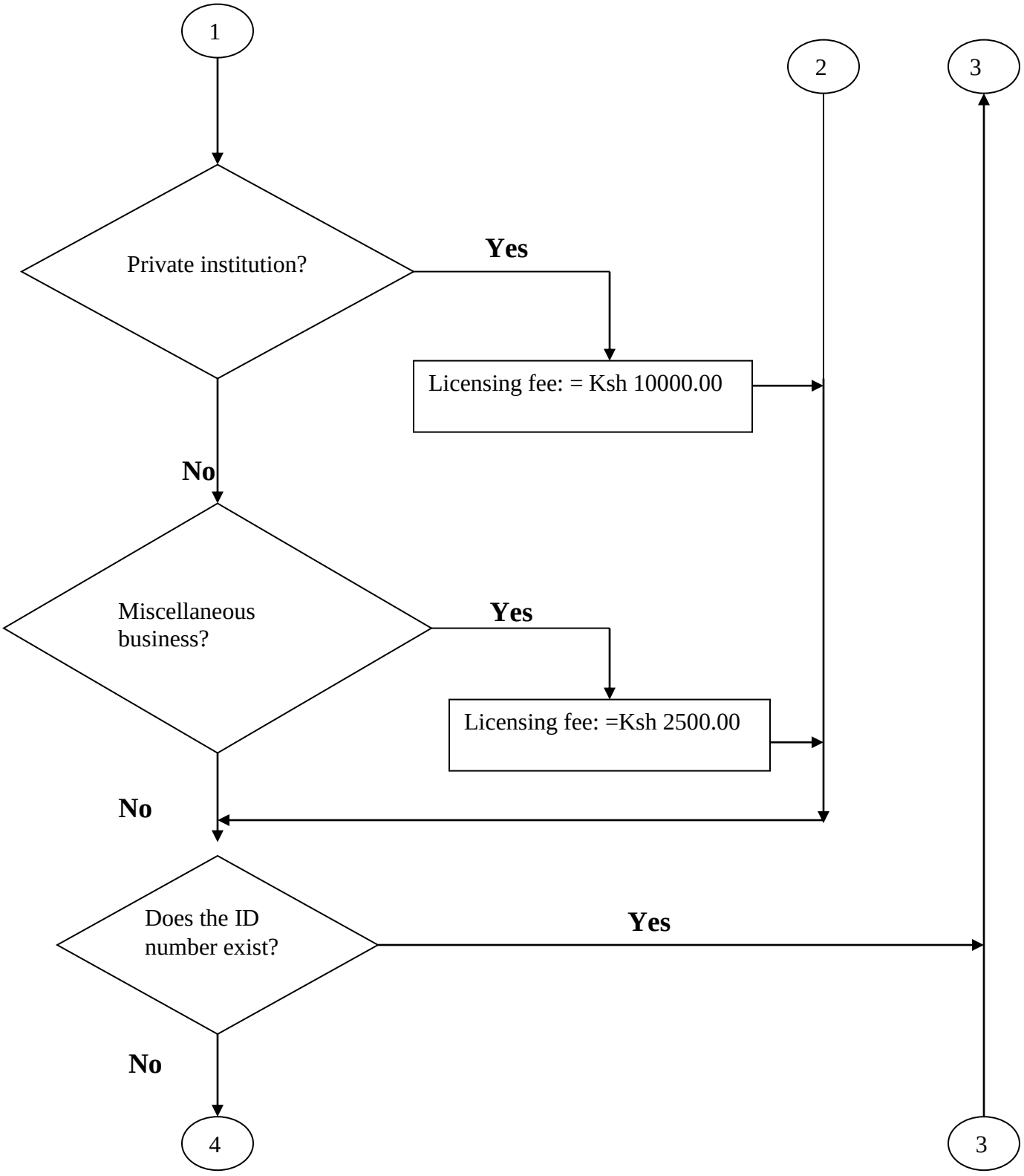
4.2.3 Procedure input resident:

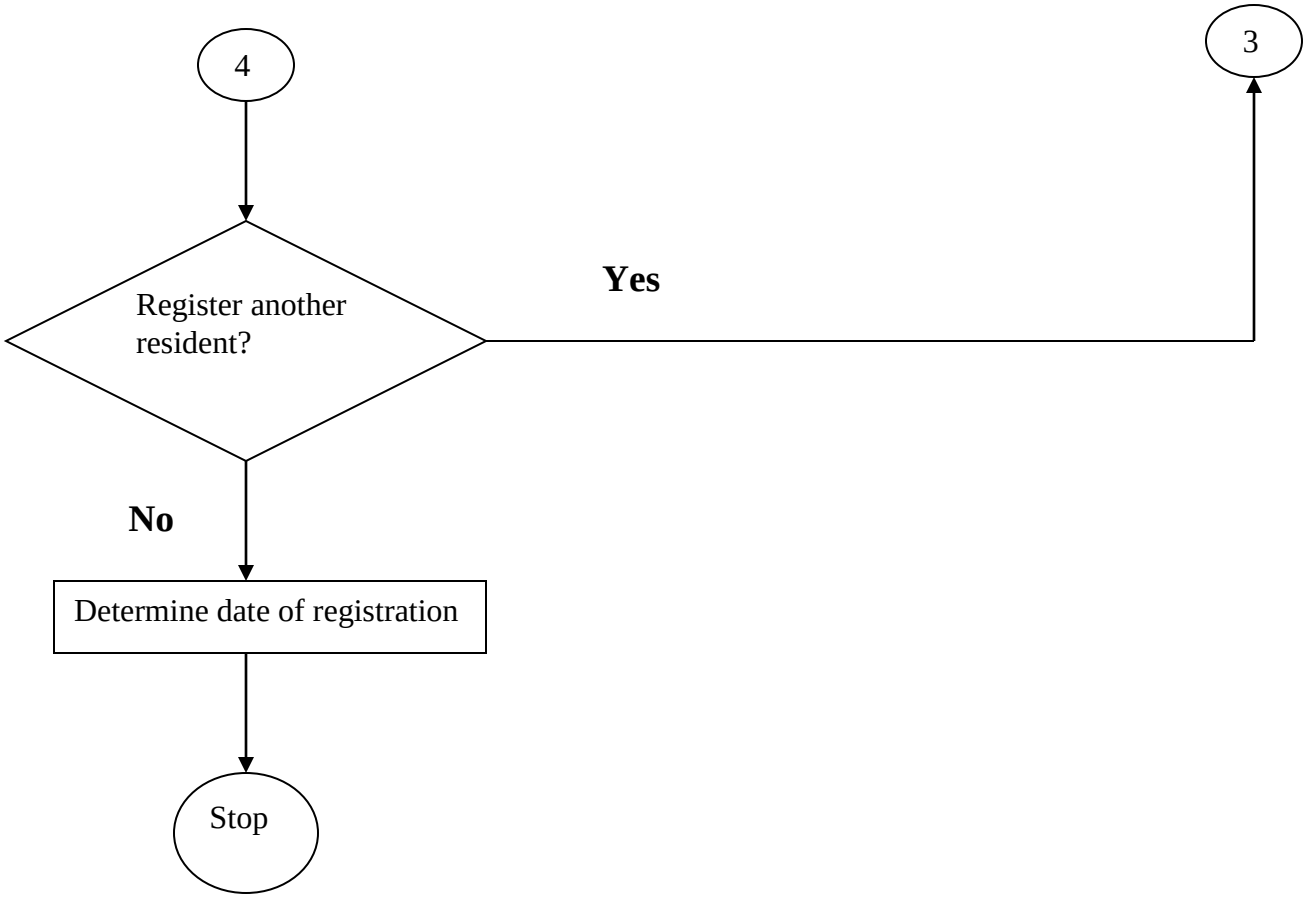



1
2
3
Three bed roomed House?
Yes
Monthly pay: = Ksh.25000
No
Four bed roomed house?
Yes
Monthly Pay: =Ksh4500
No
Does the ID number exist?
Yes
No
Register another new tenant?
Yes
No
Determine date of registration
Stop

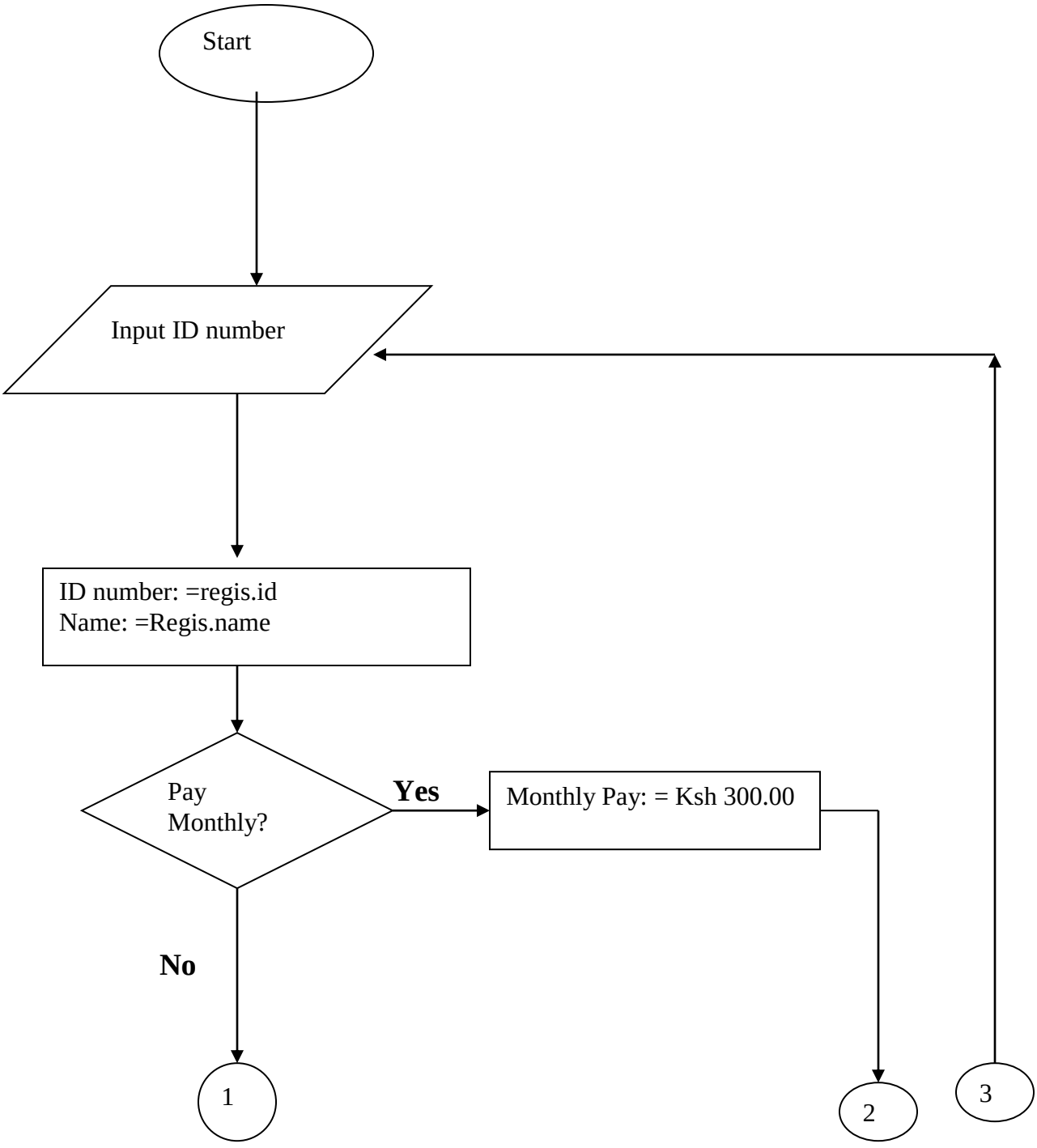
4.2.4 Procedure Input business

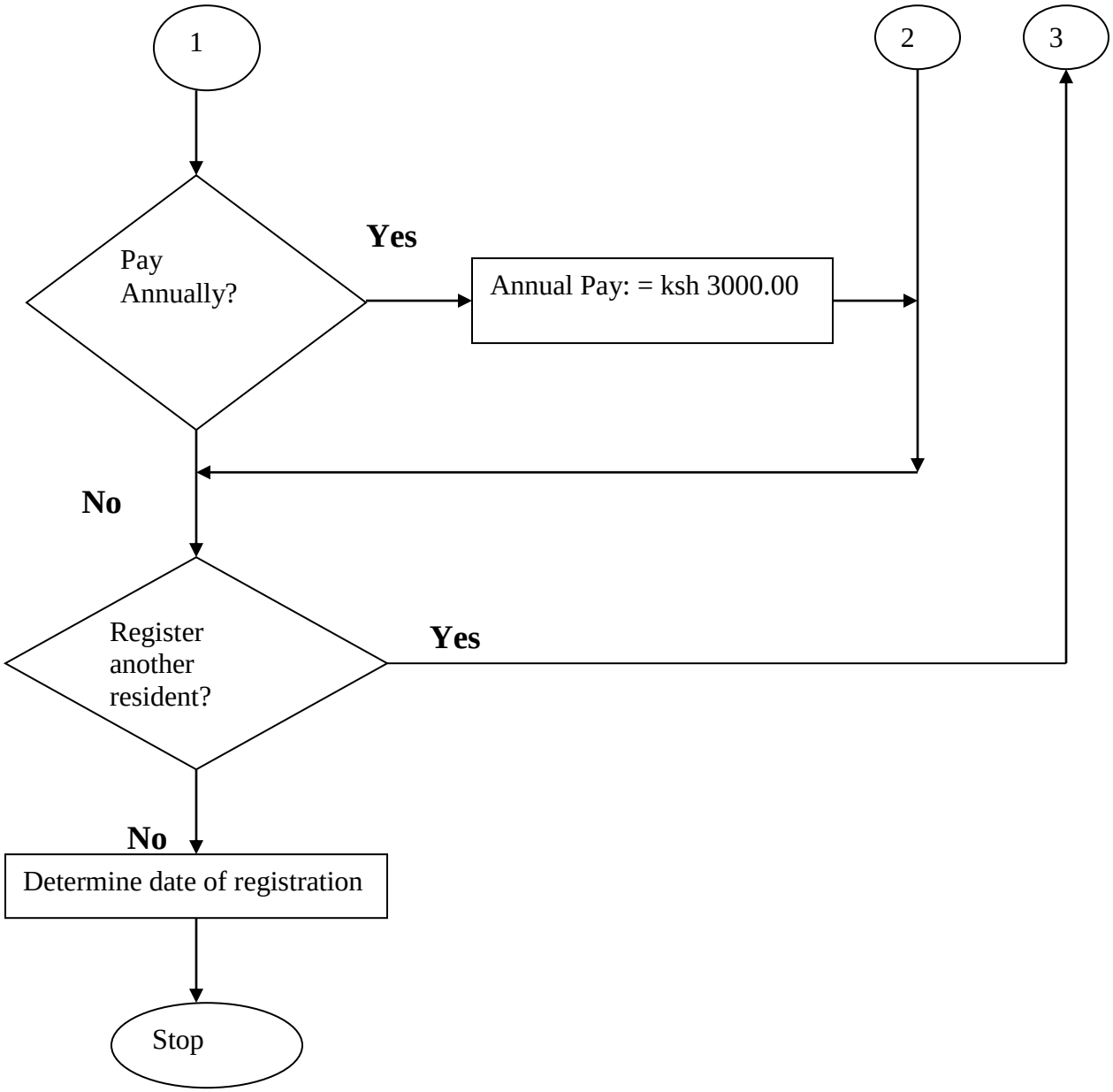



1
2
3
Private institution?
Yes
Licensing fee: = Ksh 10000.00
No
Miscellaneous business?
Yes
Licensing fee: =Ksh 2500.00
No
Does the ID number exist?
Yes
No
4
3

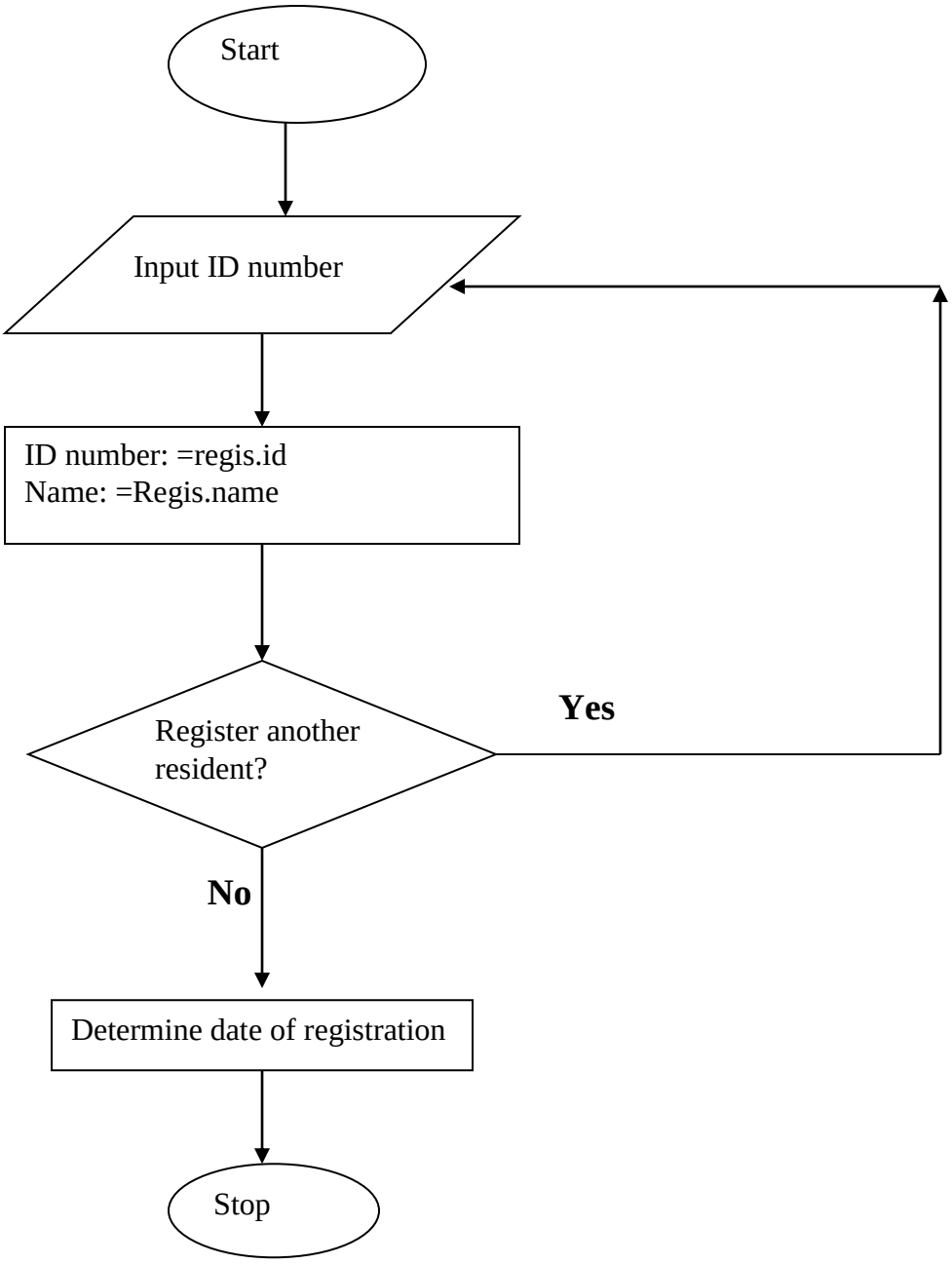

4
3
Register another resident?
Yes
No
Determine date of registration
Stop

4.2.5 Procedure input sewage

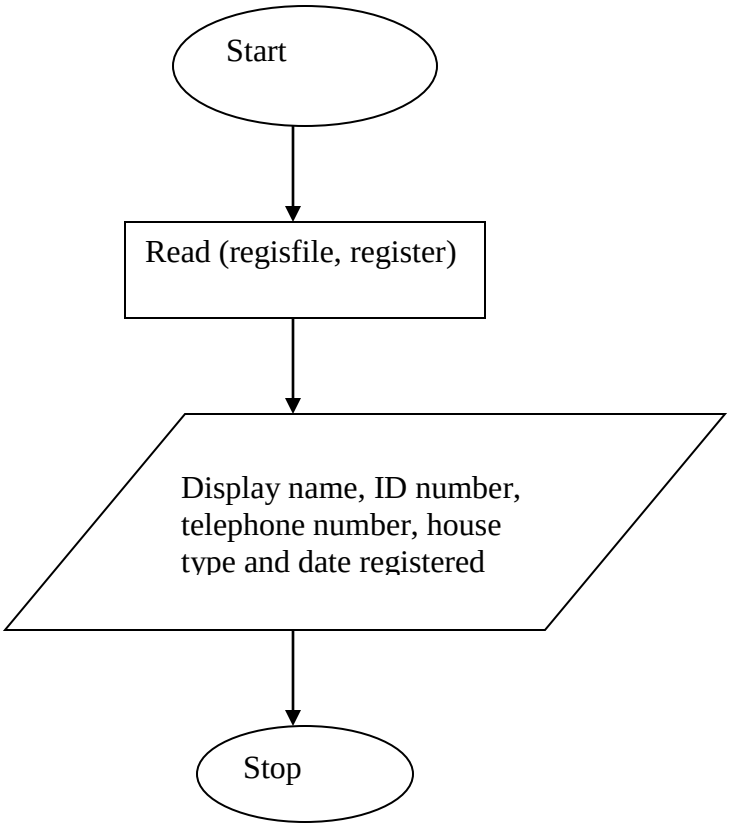



1
2
3
Pay Annually?
Yes
Annual Pay: = ksh 3000.00
No
Register another resident?
Yes
No
Determine date of registration
Stop

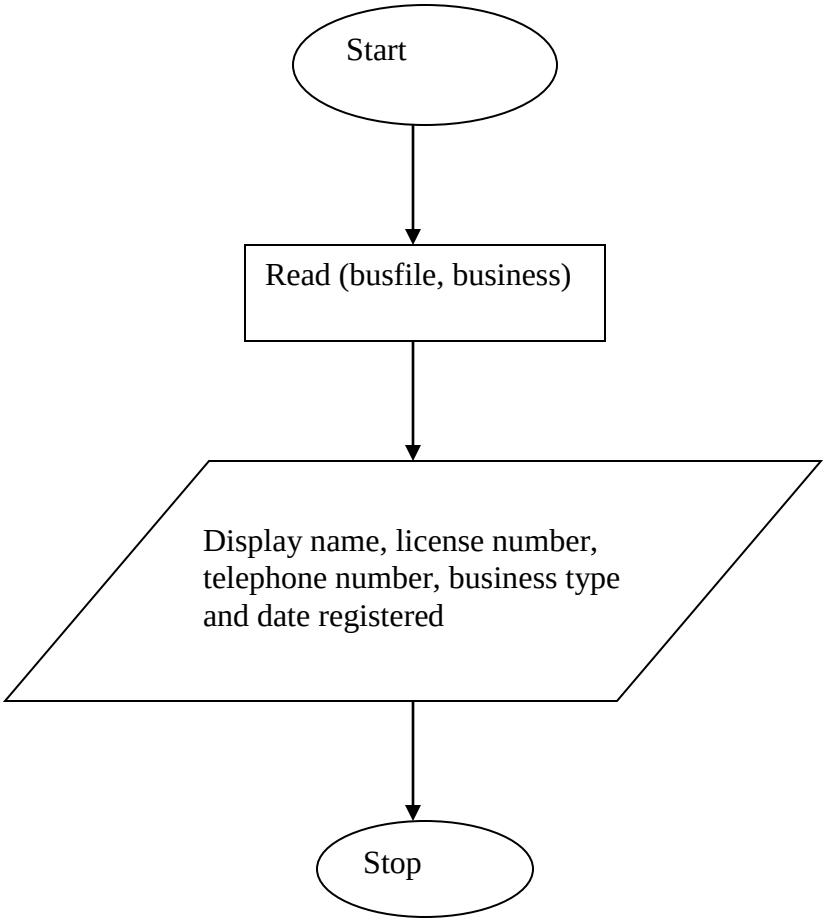
4.2.6 Procedure input garbage



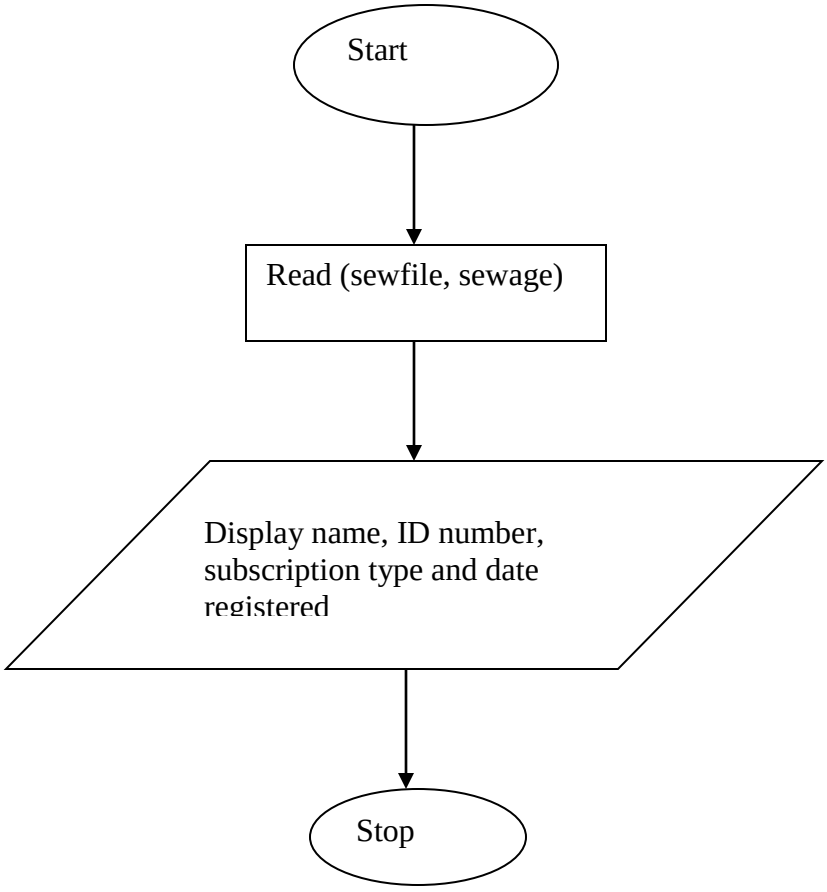
4.2.7 Procedure display tenant details



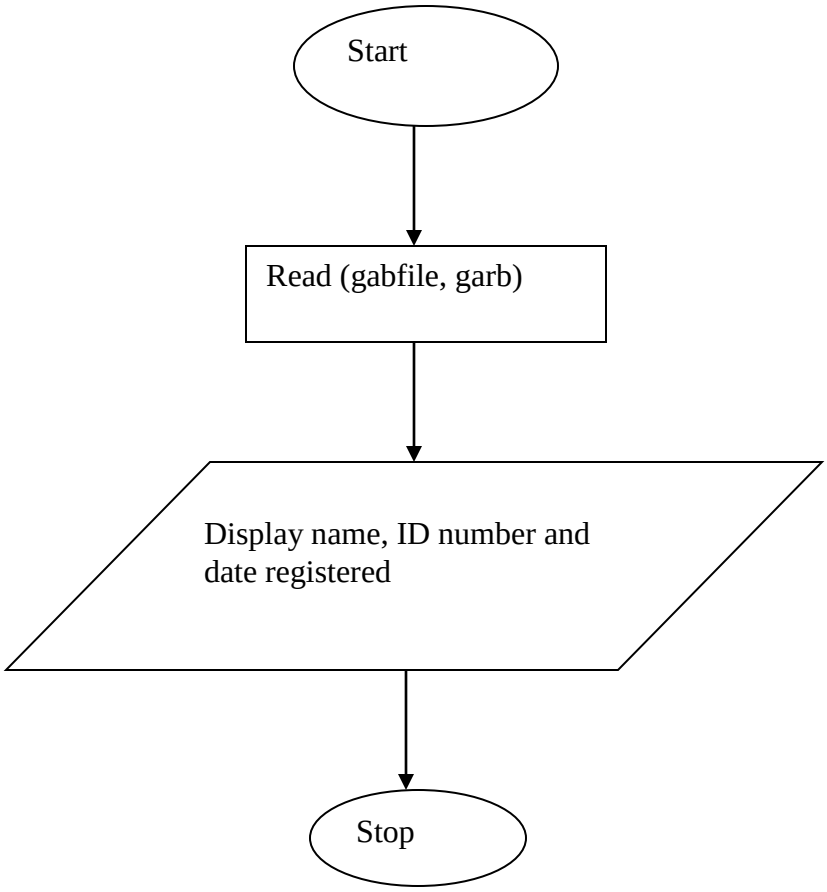
4.2.8 Procedure display business details



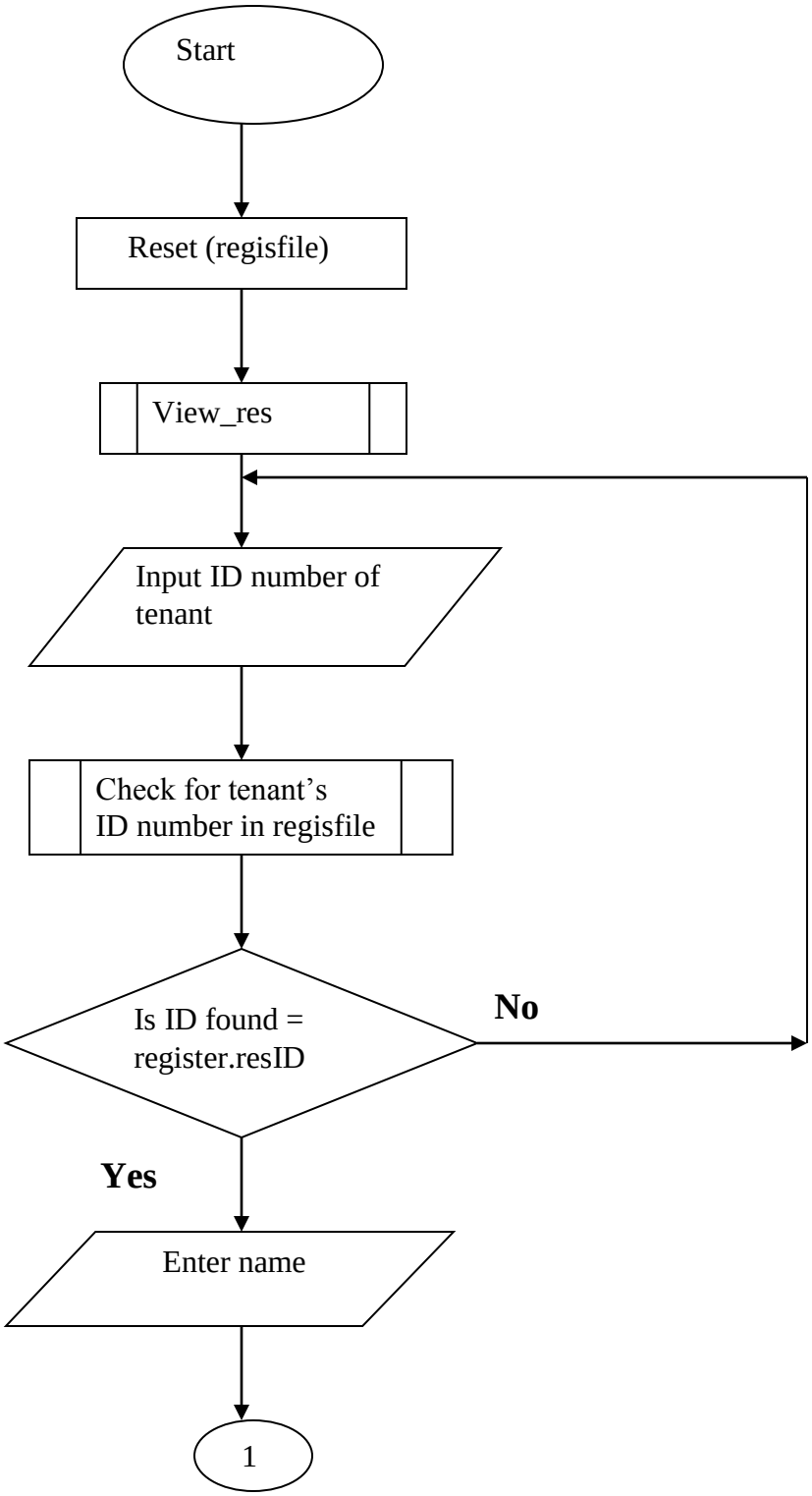
4.2.9 Procedure view sewer details

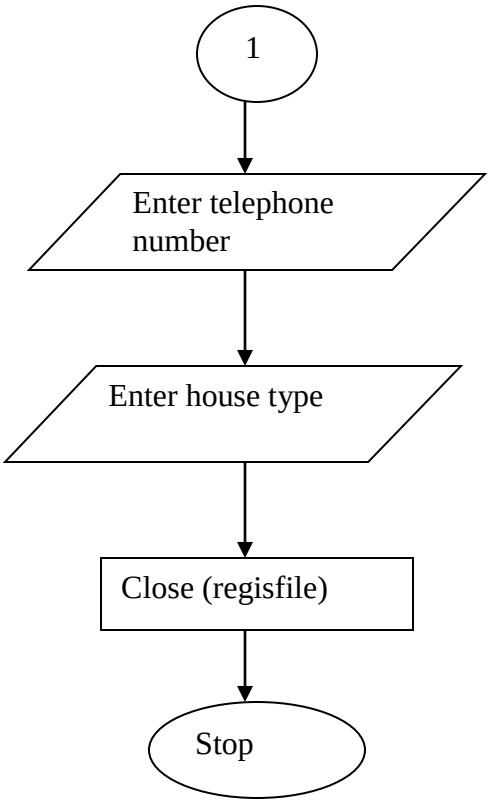


4.2.10 Procedure view garbage details

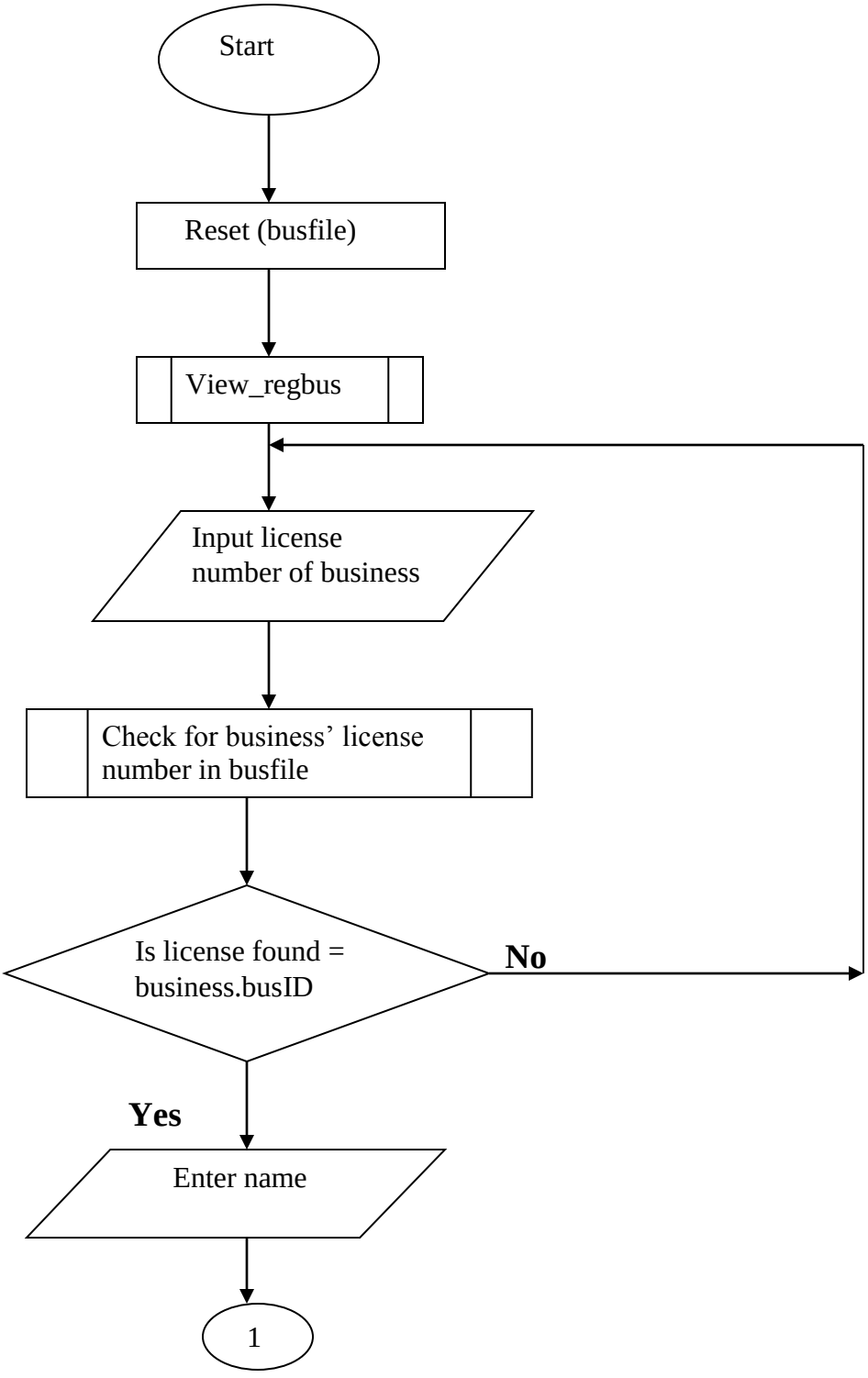


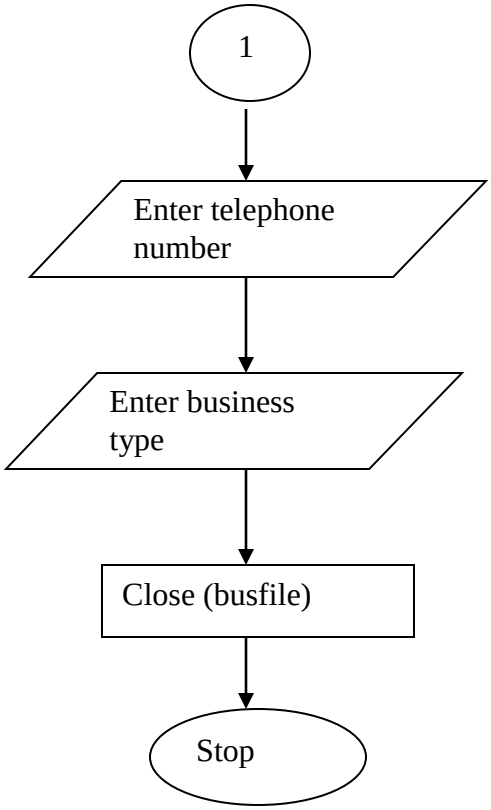
4.2.11 Procedure edit tenant details



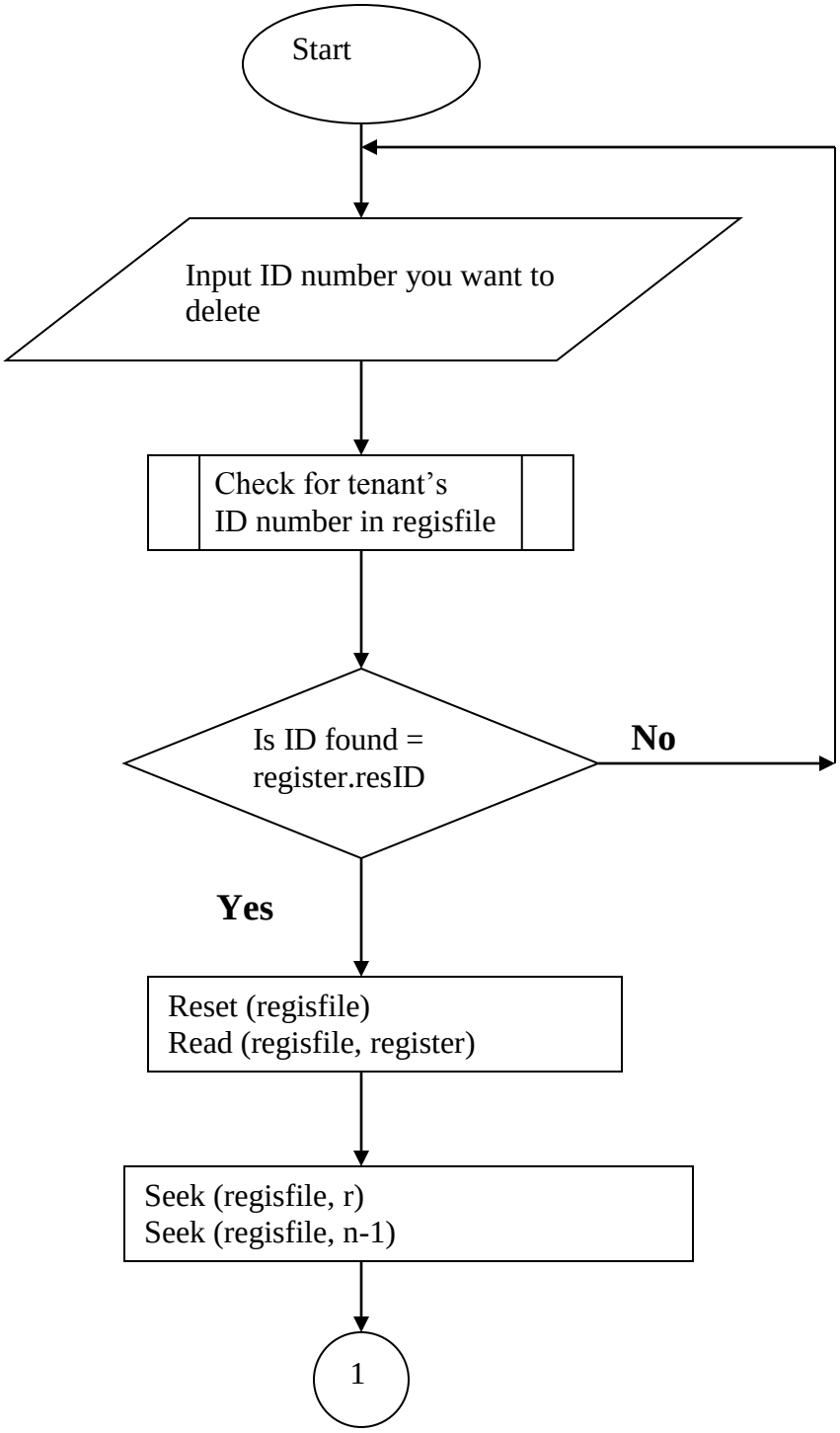

1
Enter telephone number
Enter house type
Close (regisfile)
Stop

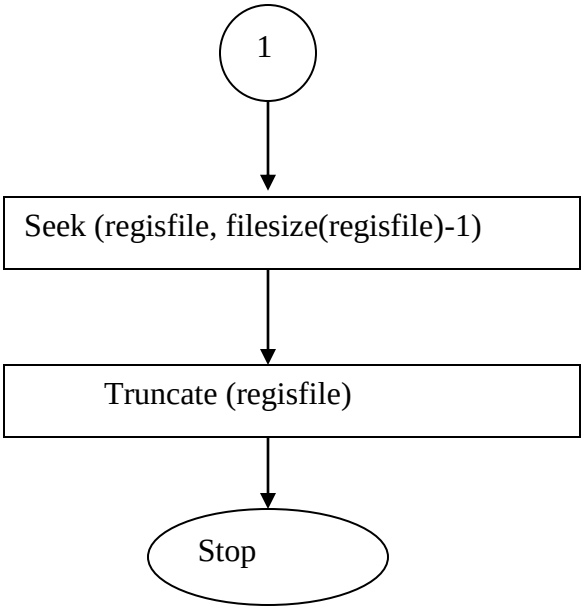
4.2.12 Procedure edit business details



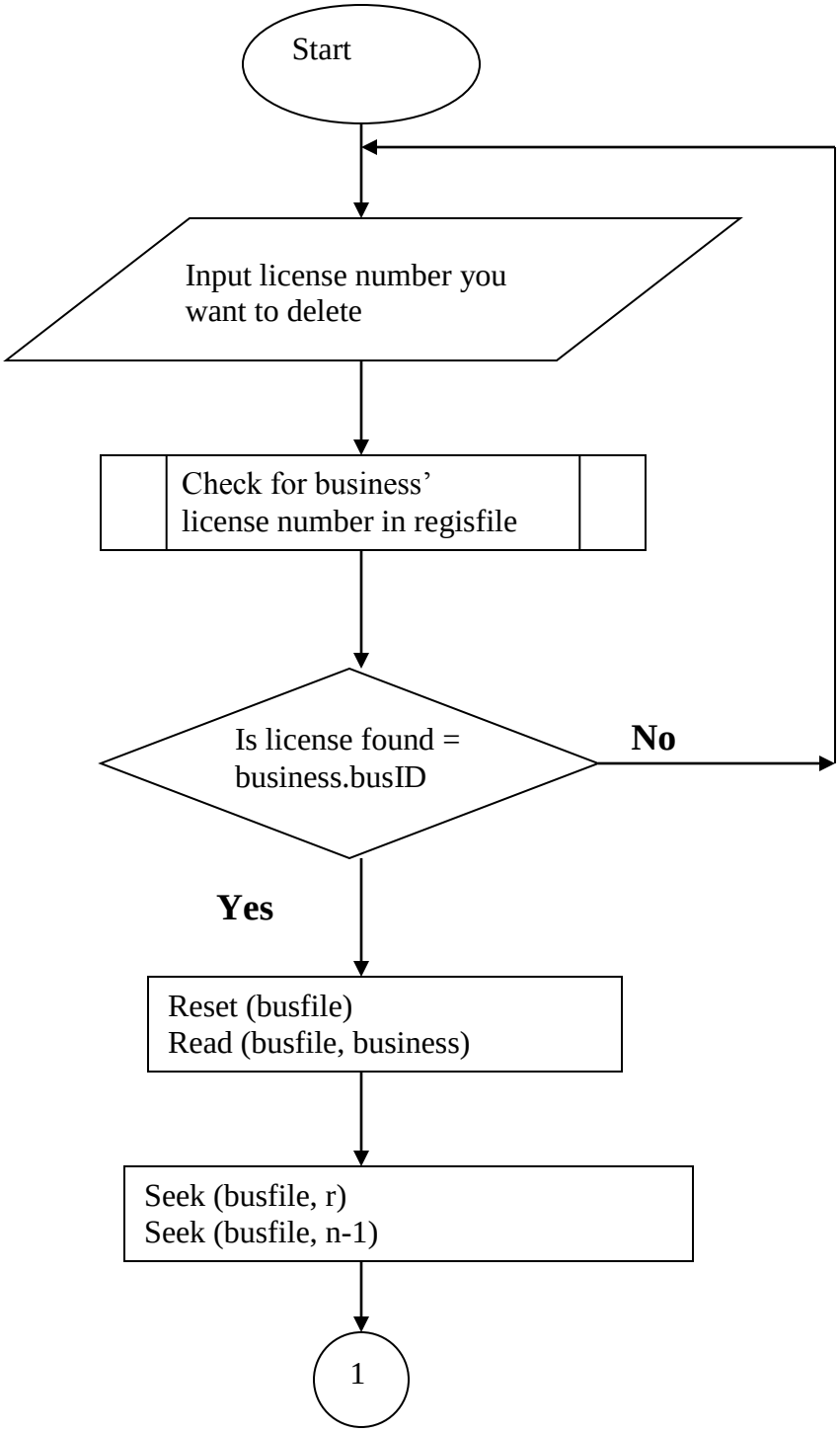

1
Enter telephone number
Enter business type
Close (busfile)
Stop

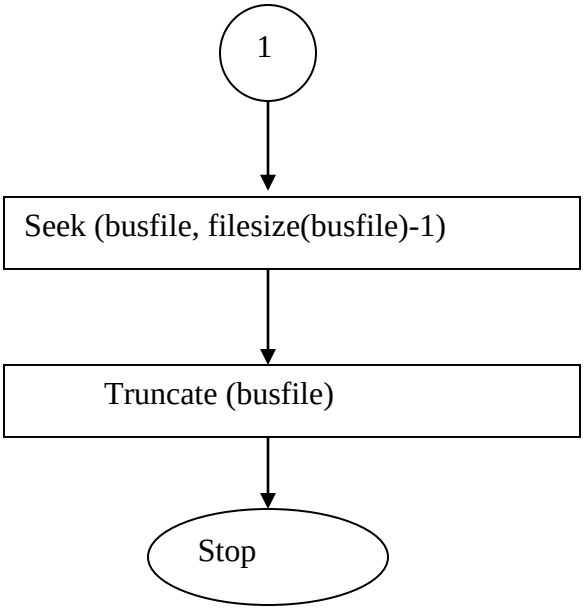
4.2.13 Procedure delete tenant



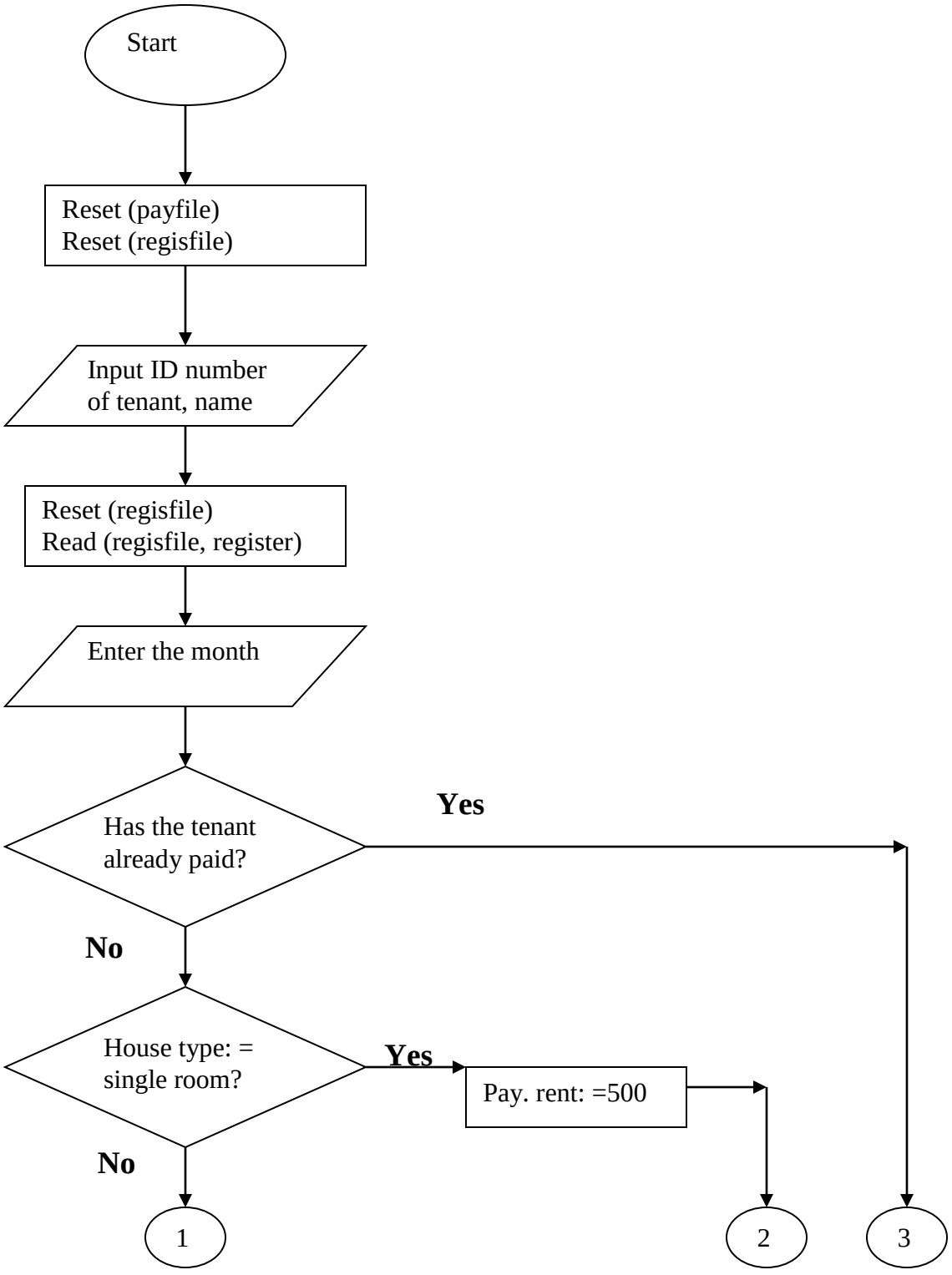

1
Seek (regisfile, filesize(regisfile)-1)
Truncate (regisfile)
Stop

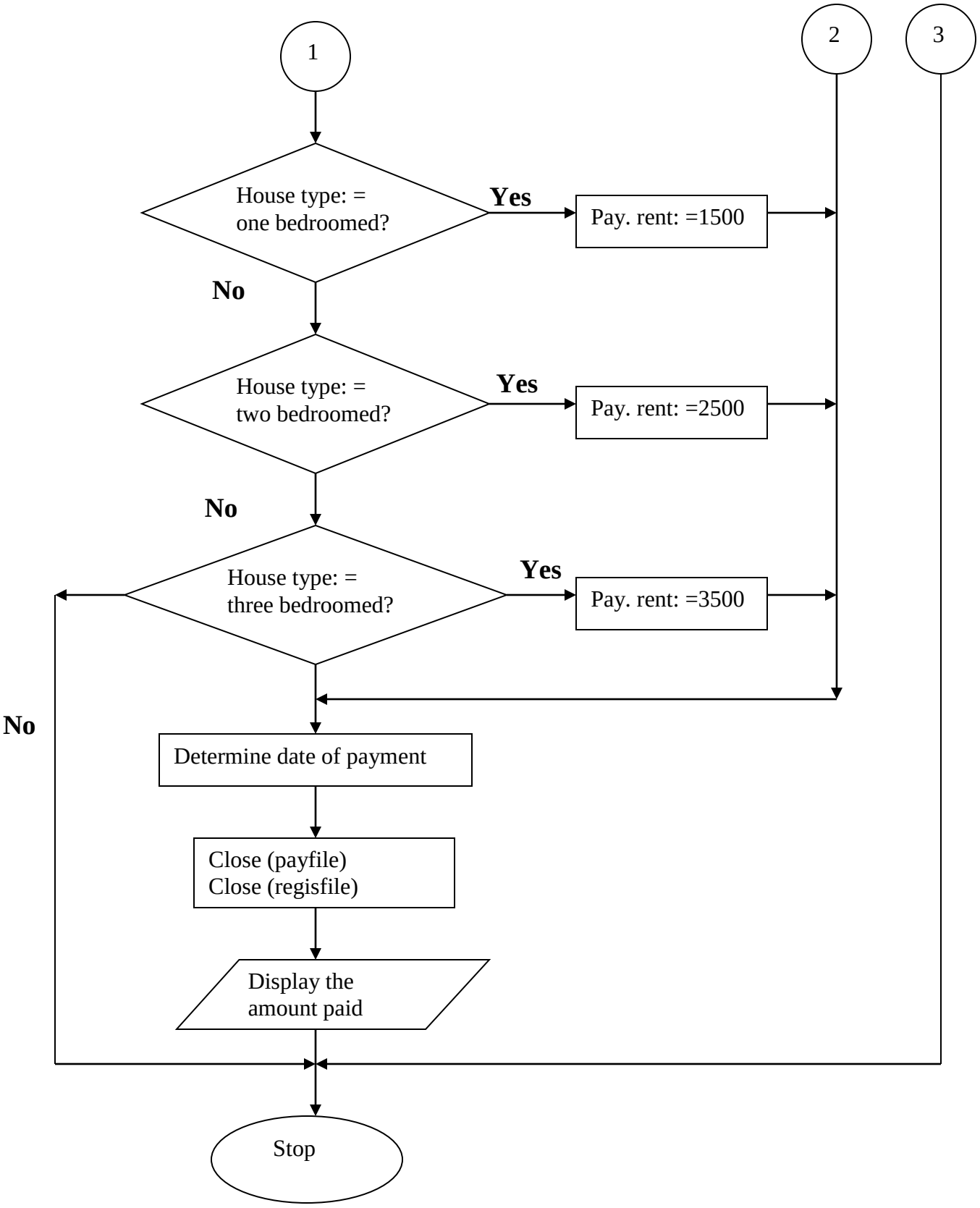
4.2.14 Procedure delete business



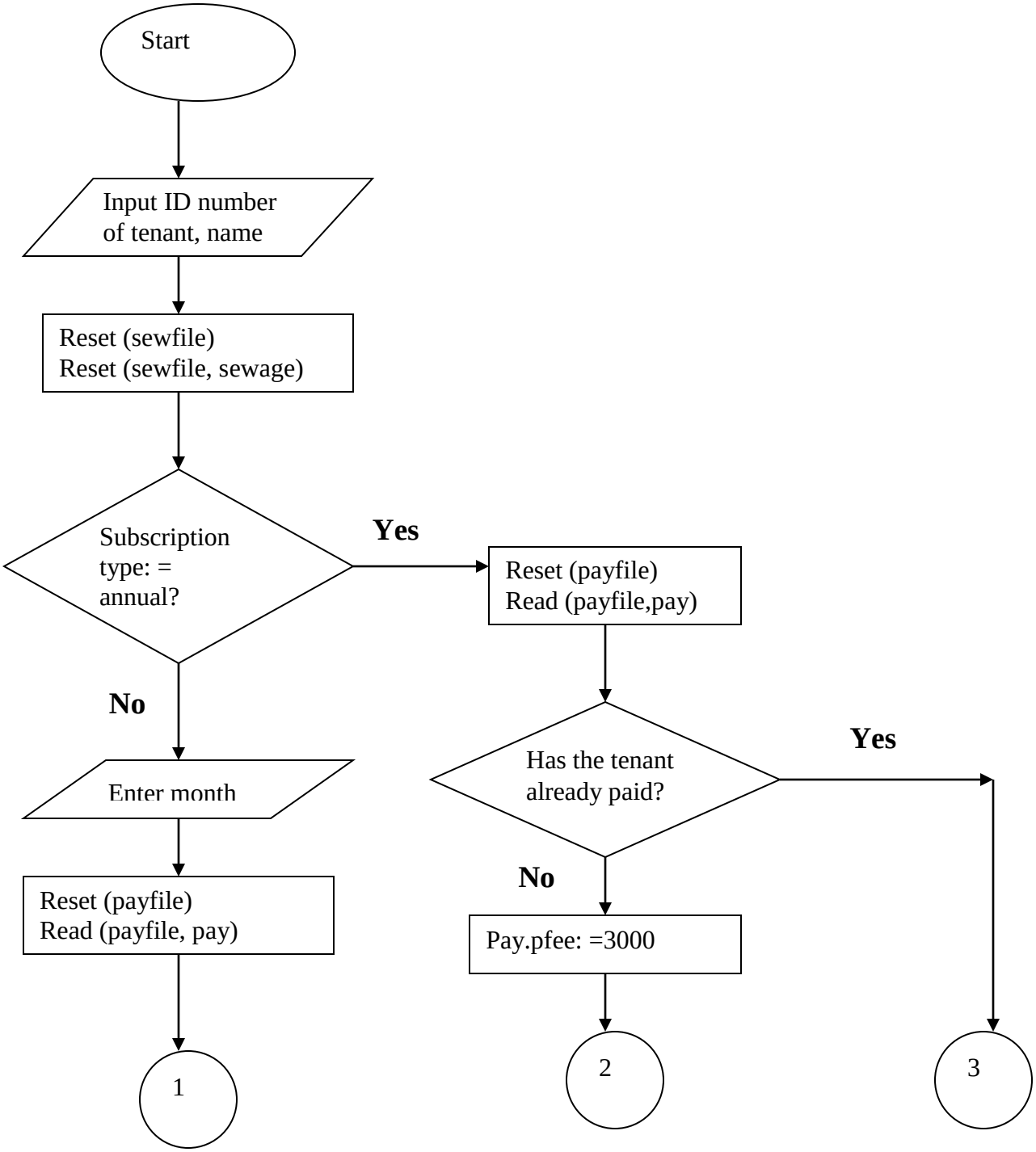

1
Seek (busfile, filesize(busfile)-1)
Truncate (busfile)
Stop

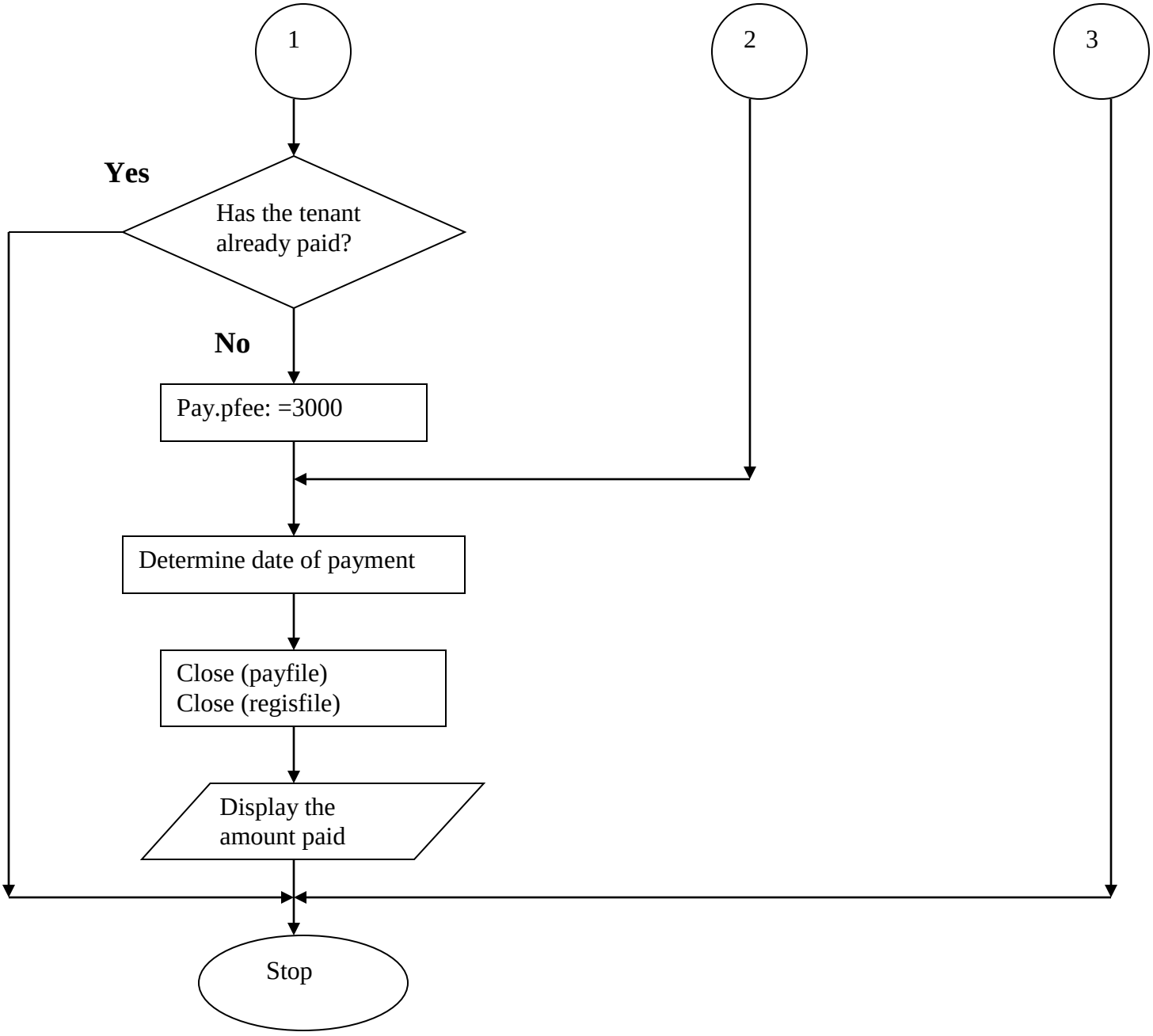
4.2.15 Procedure collect rent



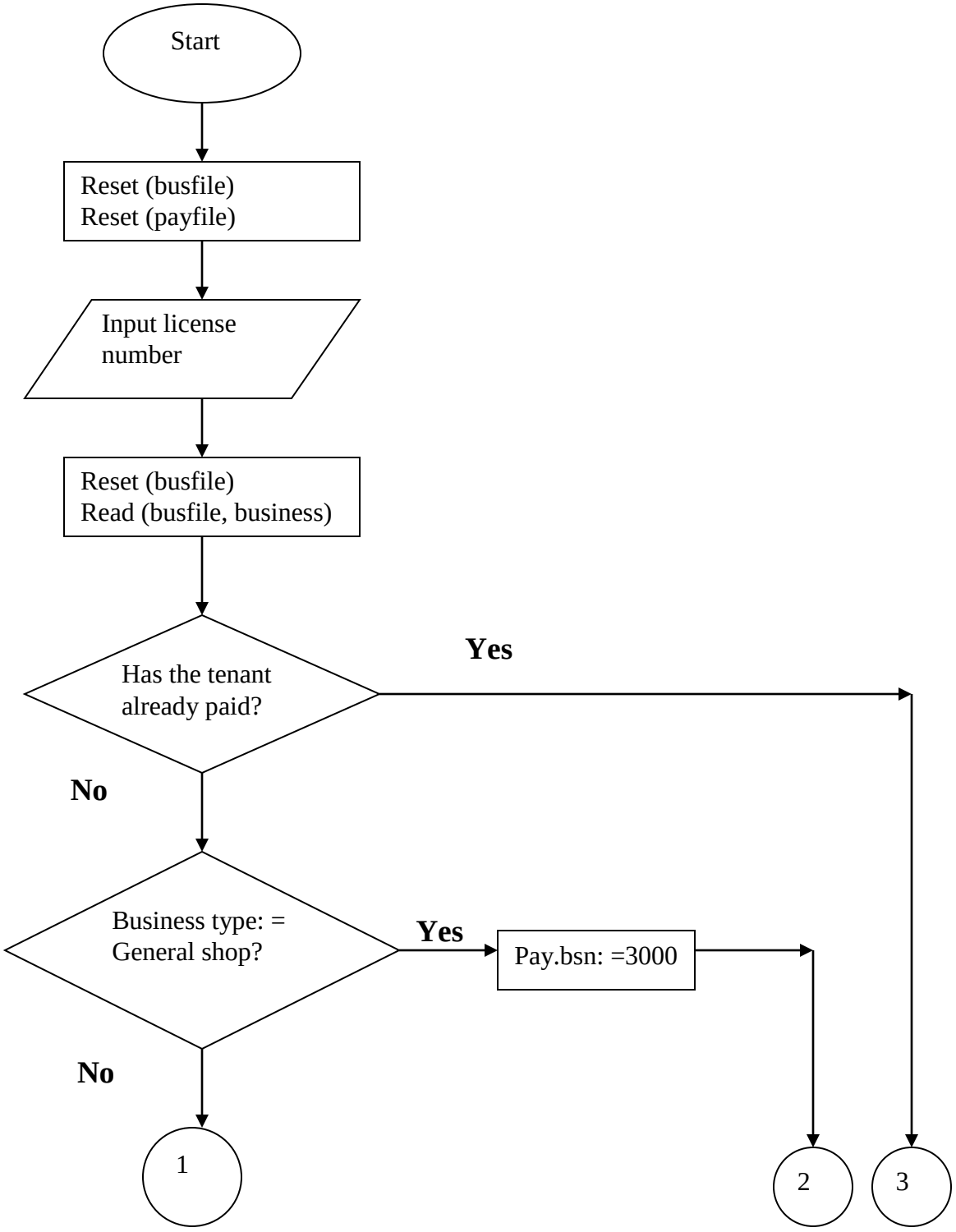

1
2
3
House type: =
one bedroomed?
Yes
Pay. rent: =1500
No
House type: =
two bedroomed?
Yes
Pay. rent: =2500
No
House type: =
three bedroomed?
Yes
Pay. rent: =3500
No
Determine date of payment
Close (payfile)
Close (regisfile)
Display the
amount paid
Stop

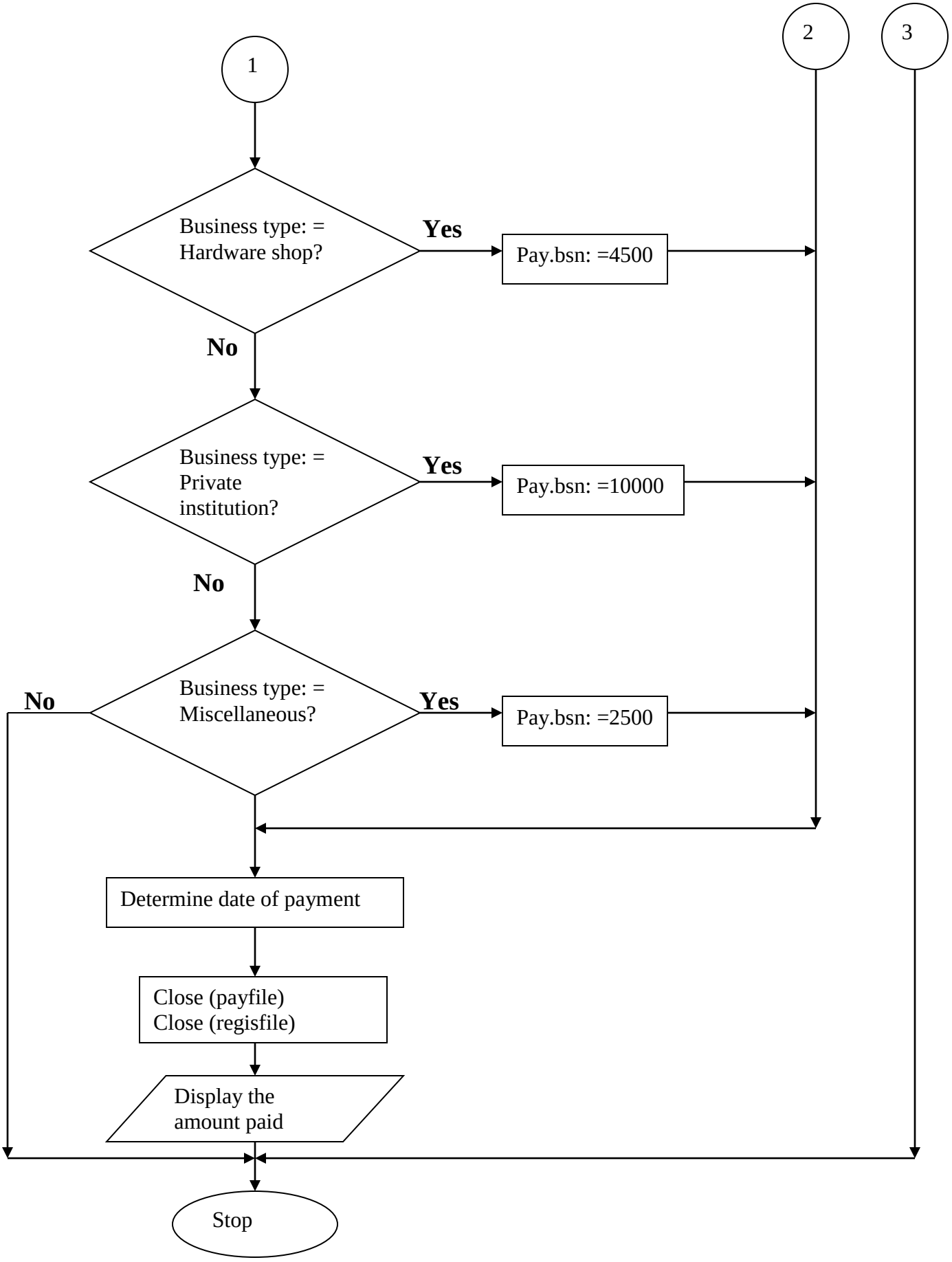
4.2.16 Procedure collect sewer payments



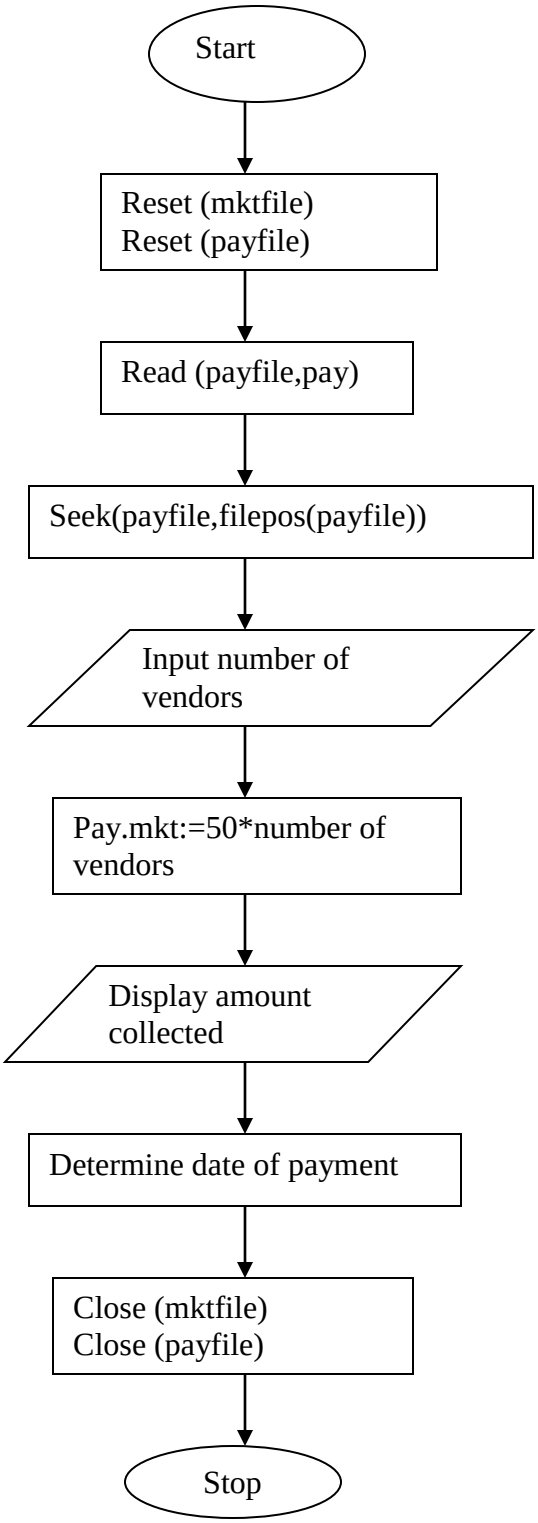

1
2
3
Yes
Has the tenant already paid?
No
Pay.pfee: =3000
Determine date of payment
Close (payfile)
Close (regisfile)
Display the amount paid
Stop

4.2.17 Procedure collect license

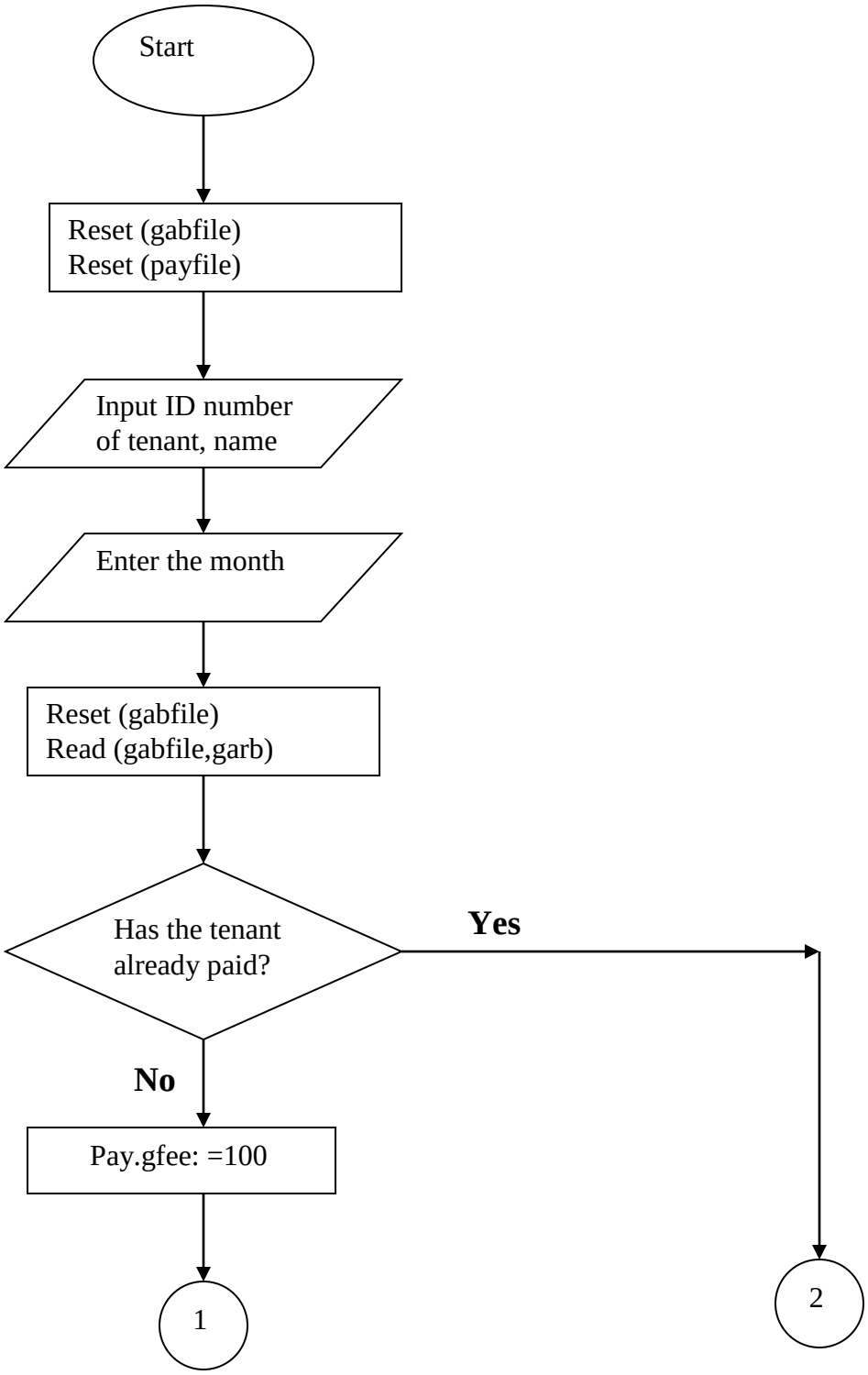


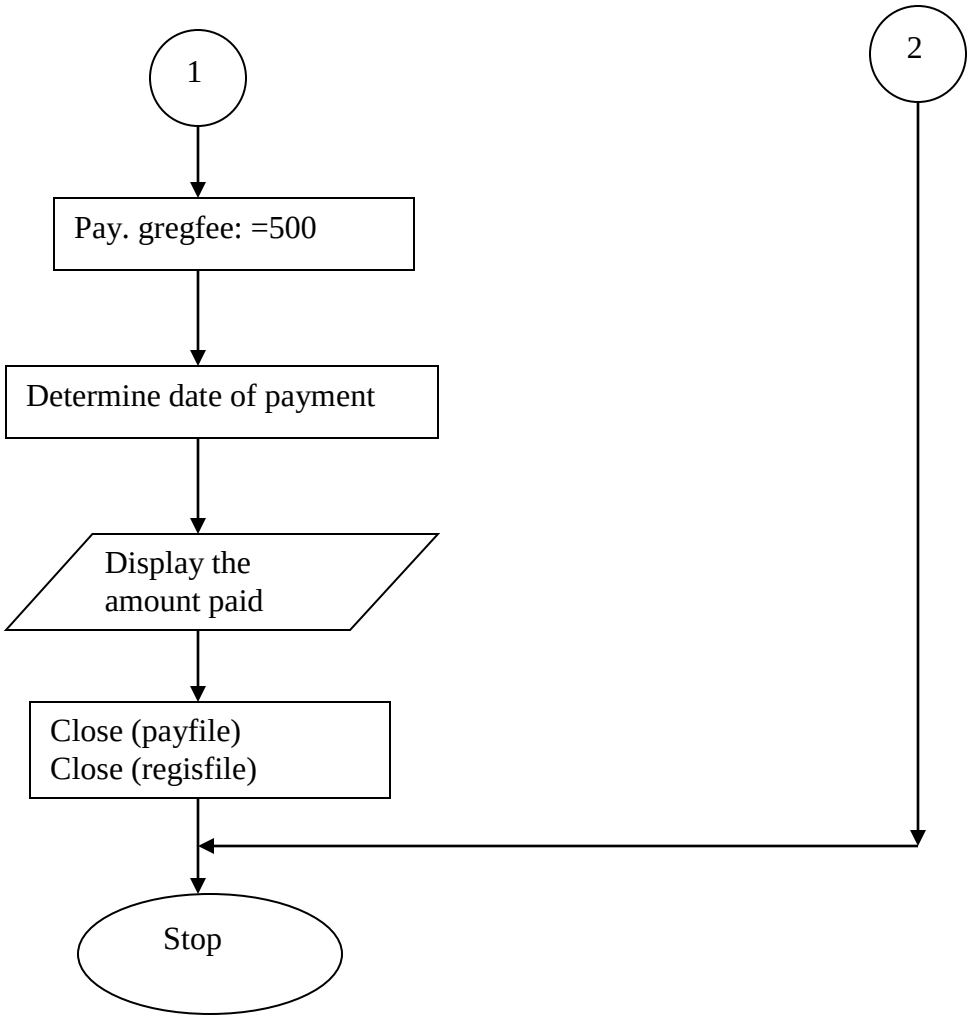

1
2
3
Business type: = Hardware shop?
Yes
Pay.bsn: =4500
No
Business type: = Private institution?
Yes
Pay.bsn: =10000
No
Business type: = Miscellaneous?
No
Yes
Pay.bsn: =2500
Determine date of payment
Close (payfile)
Close (regisfile)
Display the amount paid
Stop

4.2.18 Procedure collect market payments

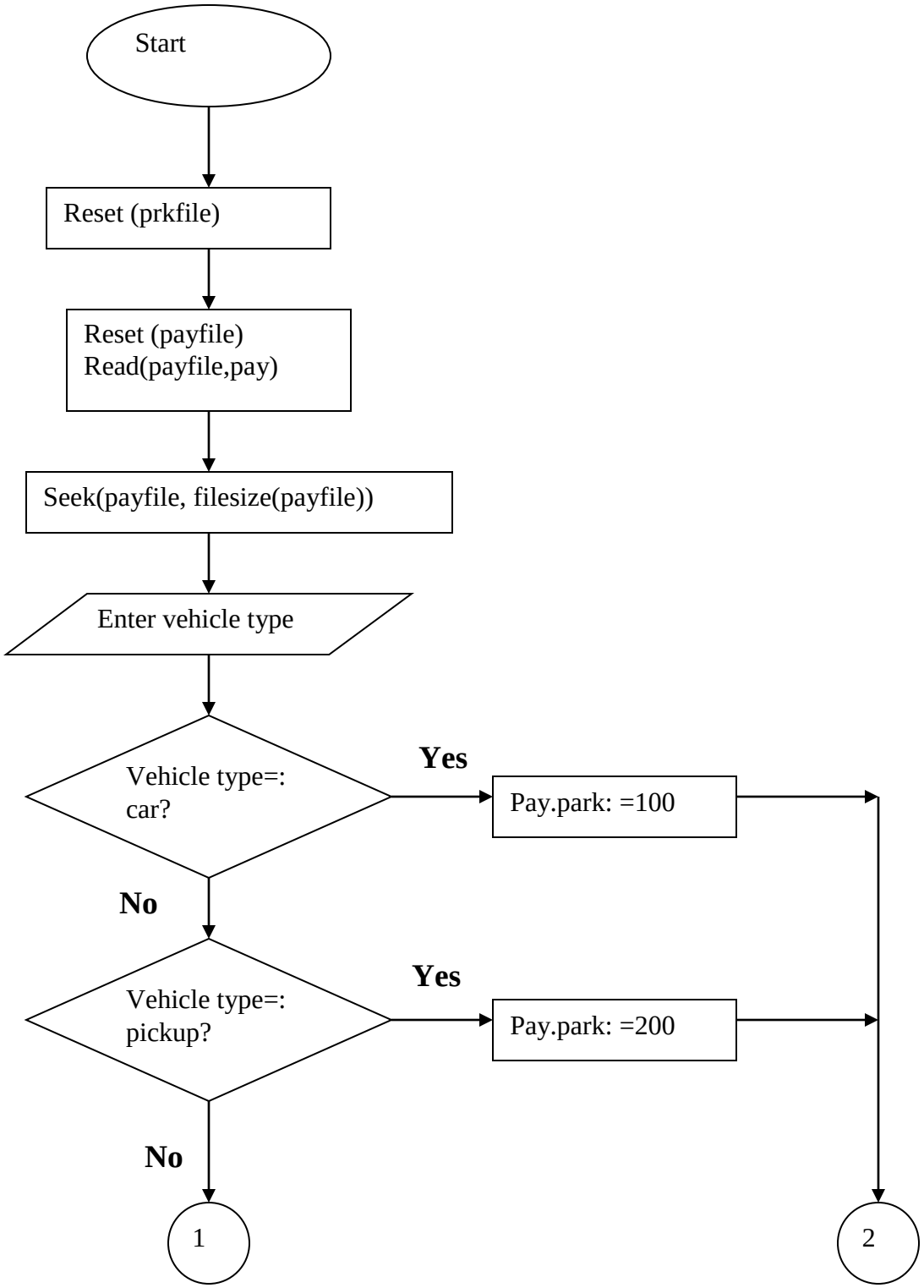


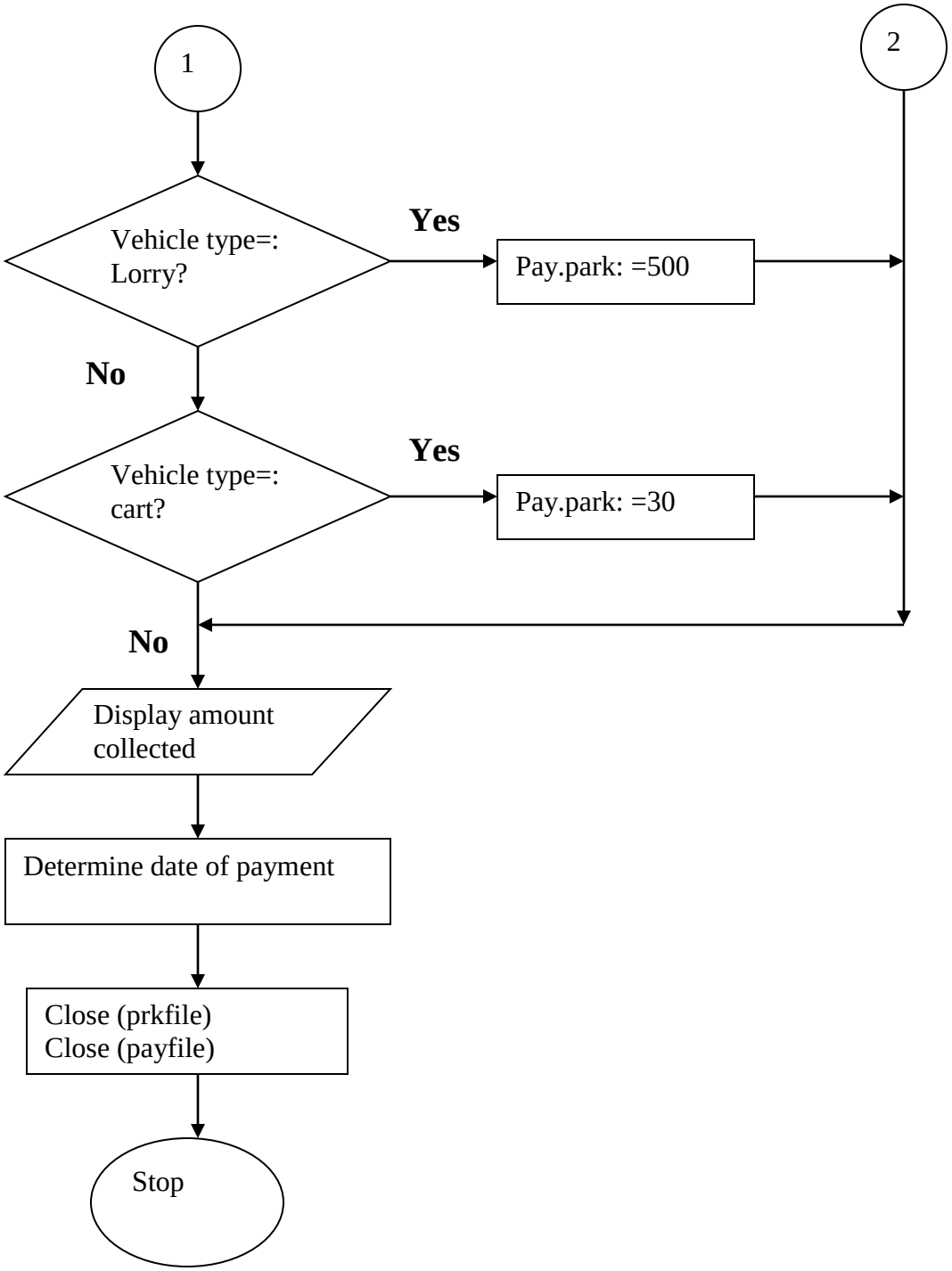
4.2.19 Procedure collect garbage



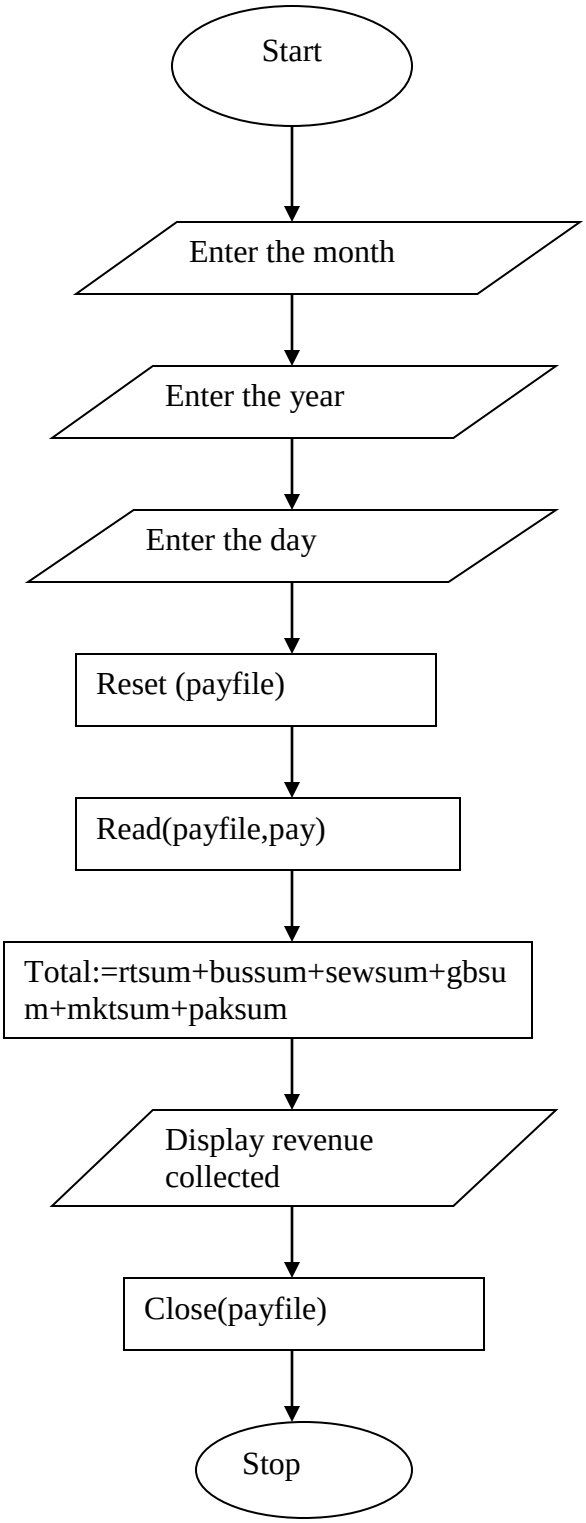

1
2
Pay. gregfee: =500
Determine date of payment
Display the
amount paid
Close (payfile)
Close (regisfile)
Stop

4.2.20 Procedure collect parking fees

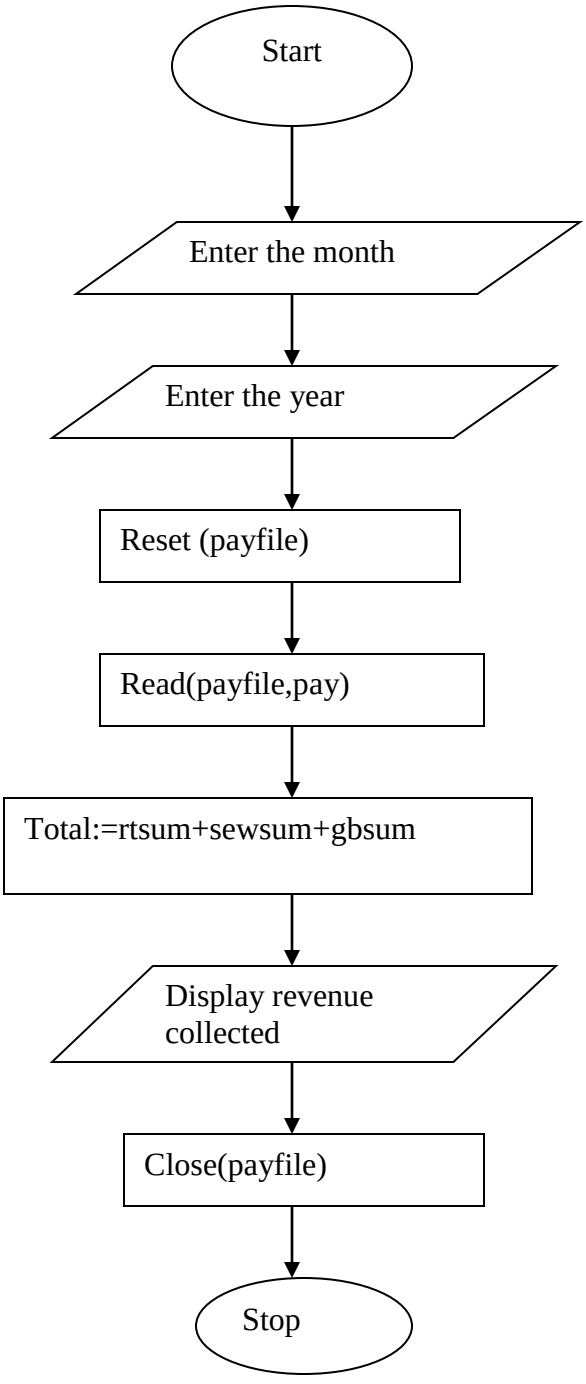



1
2
Vehicle type=:
Lorry?
Yes
Pay.park: =500
No
Vehicle type=:
cart?
Yes
Pay.park: =30
No
Display amount
collected
Determine date of payment
Close (prkfile)
Close (payfile)
Stop

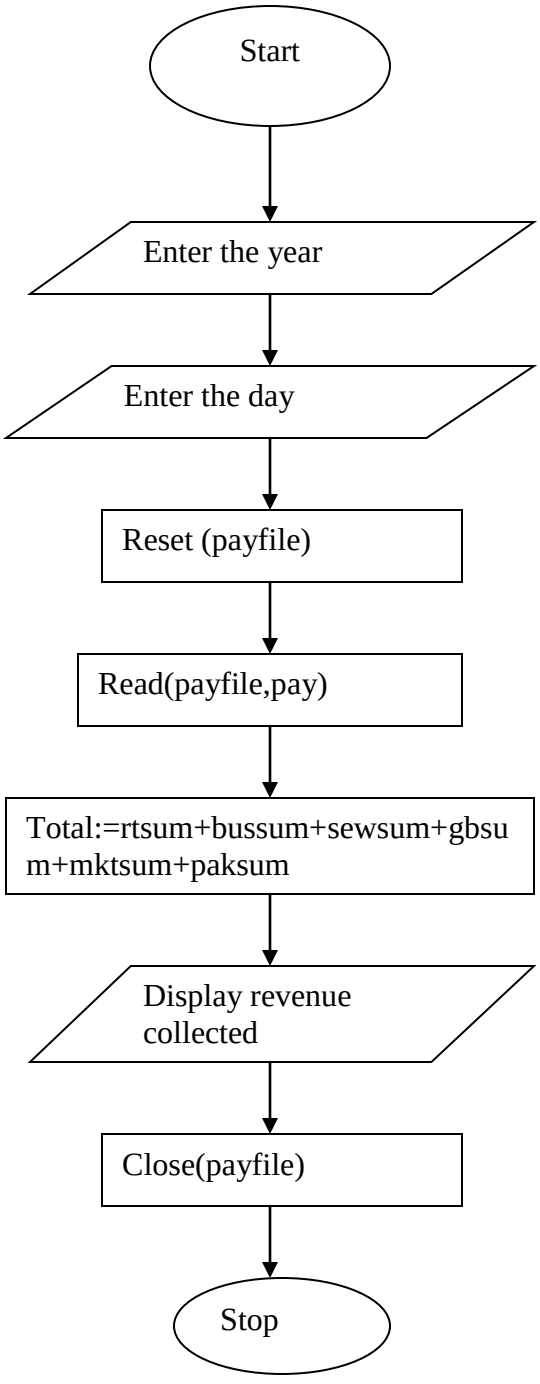
4.2.21 Procedure Daily reports



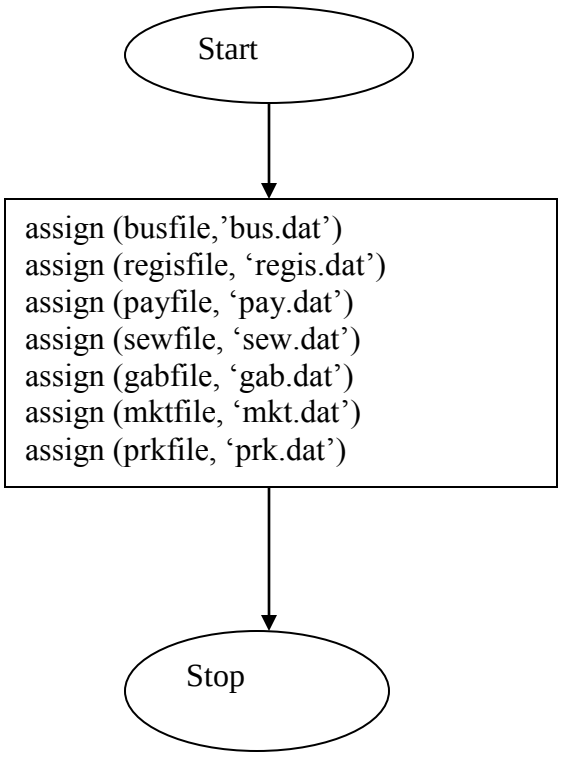
4.2.22 Procedure Monthly reports



4.2.23 Procedure Annual reports



4.2.24 Procedure assigning



5. CODING

```
PROGRAM Babati;
uses crt,dos,graph;
const
 day:array[0..6] of string=
('Sunday','Monday','Tuesday','Wednesday','Thursday','Friday
','Saturday');
 month:array[1..12] of
string=('January','February','March','April','May','June',

'July','August','September','October','November','December'
);

Type
date=record   {record containing details for the date}
     day:integer;
     month:integer;
   year:integer;
end;
Regis=record {record containing tenant details}
fname,hsetype:string;
resID:real;
tel:real;
dtreg:date;
end;
Bus=record {record containing business details}
name,bstype:string;
busID:real;
tel:real;
dtereg:date;
end;
Sewer=record  {record containing residents registered to
sewer line}
name,subtype:string;
ID:real;
datreg:date;
end;
Garbage=record  {record containing residents registered to
garbage collection}
name:string;
id:real;
dtgreg:date;
end;
payment=record {record of payments}
rent,gregfee,gfee,bsn,mkt,pfee,cfee,park,resID,busID:real;
m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12:string;
mg1,mg2,mg3,mg4,mg5,mg6,mg7,mg8,mg9,mg10,mg11,mg12:string;
```

```
ms1,ms2,ms3,ms4,ms5,ms6,ms7,ms8,ms9,ms10,ms11,ms12:string;
annual:string;
annualb:string;
dtpay:date;
monthpaid:string;
end;
Market=record {record containing market details}
vnd,total:real;
end;
Parking=record {record containing parking details}
vhctype:string;
vcamount:real;
dtpark:date;
end;
lic=record{record of licensed businesses}
name,bstype:string;
end;
VAR
regisfile:file of regis;
payfile:file of payment;
busfile:file of bus;
mktfile:file of market;
prkfile:file of parking;
sewfile:file of sewer;
gabfile:file of garbage;
tsk:integer;
register:regis;
business:bus;
pay:payment;
mkt:market;
park:parking;
sewage:sewer;
garb:garbage;
c:char;
dd,mm,yyyy,dow:word;
Procedure entry;       {procedure for starting of program}
var
  Gd, Gm, I, Width: Integer;
begin
  Gd := Detect; InitGraph(Gd, Gm, '');
  if GraphResult <> grOk then Halt(1);
    setbkcolor(blue);
    setfillstyle(10,blue);
    Bar(0, 0, 650, 600);
    setcolor(yellow);
    settextstyle(4,0,4);
    outtextxy(115,25,'BABATI URBAN COUNCIL');
```

```
    outtextxy(60,70,'REVENUE COLLECTION SYSTEM');
    setcolor(lightred);
    outtextxy(200,200,'WELCOME');
    settextstyle(5,0,3);
    setcolor(white);
    outtextxy(180,300,'Program coded by:');
    outtextxy(170,330,'Birir Lee Kipngetich');
    outtextxy(200,370,'401099114');
    settextstyle(7,0,3);
    setcolor(7);
    outtextxy(100,410,'Press <enter> key to continue');
  ReadLn;
  CloseGraph;
end;

procedure dt;     {procedure for getting date}
begin
getdate(yyyy,mm,dd,dow);
end;

Procedure menu;   {procedure for main menu}
begin;
dt;
textbackground(blue);
clrscr;
textcolor(lightred);
gotoxy(5,2);writeln('B A B A T I  R E V E N U E  C O L L E
C T I O N  S Y S T E M');
gotoxy(5,3);writeln('-------------------------------------
----------------------');
textcolor(yellow);
gotoxy(24,5);writeln('□□□□□□□ MAIN MENU □□□□□□□');
textcolor(lightgreen);
gotoxy(48,7);writeln('The current date is
',dd,'/',mm,'/',yyyy);
textcolor(white);
gotoxy(7,9);writeln('Please select a task that you want to
do:');
writeln;
writeln('       1.Registration');
writeln('       2.Collect revenue');
writeln('       3.Display  records');
writeln('       4.View reports');
writeln('       5.Edit records');
writeln('       6.Delete records');
writeln('       7.Exit program');
readln(tsk);
```

```
 if tsk>7 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(tsk);
       until tsk<=7
       end;


Procedure input_res;    {procedure for registering tenants}
VAR
hs,ndai:integer;
id,v:real;
n,s:string;
begin
       reset(regisfile);
    while not eof(regisfile)  do
    read(regisfile,register);

     reset(payfile);
    while not eof(payfile)  do
    read(payfile,pay);
    repeat
    clrscr;

     writeln('Enter the name of the resident');readln(n);
     readln(n);
     writeln('Enter the ID number of the resident');
    repeat
    reset(regisfile);
    readln(ID);V:=1;
    while not eof(regisfile) do
    begin
    read(regisfile,register);
    if register.resID=ID then
    begin
    writeln('The ID Number already exists.Enter another
valid one');
    V:=0;
    end;
    end;
    until v=1;
    register.resID:=ID;
    register.fname:=n;
    pay.resID:=register.resID;
    writeln('Enter the telephone number of the resident');
    readln(register.tel);
    writeln('Enter the type of house');
```

```
    writeln('           HOUSES:                        ');
    writeln('                    ɔ');
    writeln('    House Type            Rent         ');
    writeln(' 1Single room      Ksh.500.00     ');
    writeln(' 2One bedroomed    Ksh.1,500.00   ');
    writeln(' 3Two bedroomed    Ksh.2,500.00   ');
    writeln(' 4Three bedroomed  Ksh.3,500.00   ');
    writeln('                    ɔ');
    write('House type:');read(hs);
    if hs>4 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT HOUSE
TYPE');
       readln(hs);
       until hs<=4;
    case hs of
          1:register.hsetype:='Single room';
          2:register.hsetype:='One bedroomed';
          3:register.hsetype:='Two bedroomed';
          4:register.hsetype:='Three bedroomed';
          end;

    dt;
    writeln('The date of registration
is:',dd,'/',mm,'/',yyyy);readln;
    register.dtreg.day:=dd;
    register.dtreg.month:=mm;
    register.dtreg.year:=yyyy;
    s:='';
    pay.m1:=s;
    pay.m2:=s;
    pay.m3:=s;
    pay.m4:=s;
    pay.m5:=s;
    pay.m6:=s;
    pay.m7:=s;
    pay.m8:=s;
    pay.m9:=s;
    pay.m10:=s;
    pay.m11:=s;
    pay.m12:=S;

    seek(regisfile,filesize(regisfile));
    seek(payfile,filesize(payfile));
    write (regisfile,register);
    write(payfile,pay);
```

```
    write('DO YOU WANT TO INPUT ANOTHER TENANT (Y/N) ?');
    repeat
        c:=upcase(readkey);
    until c in ['Y','N'];
     writeln(c);
    until c='N';
   close(regisfile);
   close(payfile);

             end;

Procedure input_bus;    {procedure for registering
businesses}
Var
  bs,gari:integer;
  id,v:real;
  n,s:string;
begin
clrscr;
 dt;

       reset(busfile);
    reset(payfile);
    while not eof(payfile)  do
    read(payfile,pay);
    repeat
    clrscr;
    writeln('Enter the name of the business');
    readln(n);  readln(n);
    writeln('Enter the license number of the business');
     repeat
     reset(busfile);
    readln(ID);V:=1;
    while not eof(busfile) do
    begin
    read(busfile,business);
    if business.busID=ID then
    begin
    writeln('The ID Number already exists.Enter another
valid one');
    V:=0;
    end;
    end;
    until v=1;
    business.busID:=ID;
    business.name:=n;
```

```
    pay.busID:=business.busID;

    writeln('Enter the type of business preffered');
     writeln('       BUSINESSES:                    ');
    writeln('                   ɔ');
    writeln('  Business type       License fee     ');
    writeln(' 1:General shop         Ksh.3000.00   ');
    writeln(' 2:Hardware shop        Ksh.4500.00   ');
    writeln(' 3:Private institution  Ksh.10000.00  ');
    writeln(' 4:Miscellaneous        Ksh.2500.00   ');
    writeln('                   ɕ');
    write('Business type:');read(bs);
    if bs>4 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT BUSINESS
TYPE');
       readln(bs);
       until bs<=4;
     case bs of
          1:business.bstype:='General shop';
          2:business.bstype:='Hardware shop';
          3:business.bstype:='Private institution';
          4:business.bstype:='Miscellaneous';

          end;

    writeln('Enter the telephone no. of the business');
    readln(business.tel);
    writeln('The date of registration
is:',dd,'/',mm,'/',yyyy);
    business.dtereg.day:=dd;
    business.dtereg.month:=mm;
    business.dtereg.year:=yyyy;
    s:='';
    pay.annualb:=s;

     seek(busfile,filesize(busfile));
     seek(payfile,filesize(payfile));
     write (busfile,business);
     write(payfile,pay);
    write('DO YOU WANT TO INPUT ANOTHER BUSINESS (Y/N) ?');
    repeat
        c:=upcase(readkey);
    until c in ['Y','N'];
     writeln(c);
    until c='N';
    close(busfile);
```

```
    close(payfile);
      end;

Procedure input_sewer;  {procedure for registering sewer
connection}
var
sb:integer;
id:real;
begin
dt;
clrscr;
     reset(sewfile);
    while not eof(sewfile)  do
    read(sewfile,sewage);
    repeat
    clrscr;

writeln('Residents pay the following to be connected:');
writeln('                    ↄ ');
writeln(' 1.Initial connection fee  - Ksh 3500  ');
writeln(' 2.Annual Subcription fee  - Ksh 3000  ');
writeln(' 3.Monthly Subcription fee - Ksh 300   ');
writeln('                    ↄ ');
writeln;textcolor(white);
Writeln('Press any key to continue......');readkey;

clrscr;
writeln('Enter the ID number of the tenant who');
write('wants to be connected:');
readln(id);
reset(regisfile);
repeat
read(regisfile,register)
until register.resid=id;
     writeln('Which subscription does the resident want to
connect to?');
     writeln('1.Annual subscription');
     writeln('2.Monthly subscription');
     write('Subscription:');readln(sb);
     if sb>2 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT
SUBSCRIPTION TYPE');
       readln(sb);
       until sb<=2;
     case sb of
      1:sewage.subtype:='Annual subscription';
```

```
      2:sewage.subtype:='Monthly subscription';
      end;
      writeln('The date of registration
is:',dd,'/',mm,'/',yyyy);
      sewage.datreg.day:=dd;
     sewage.datreg.month:=mm;
     sewage.datreg.year:=yyyy;
     sewage.name:=register.fname;
     sewage.id:=register.resID;

      seek(sewfile,filesize(sewfile));
      write(sewfile,sewage);
    writeln('DO YOU WANT TO REGISTER ANOTHER
RESIDENT?(Y/N)');
      repeat
        c:=upcase(readkey);
    until c in ['Y','N'];
     writeln(c);
    until c='N';
    close(sewfile);
      end;

Procedure input_garbage;    {procedure for registering
garbage collection}
var id:real;
begin
dt;
clrscr;

       reset(gabfile);
       reset(payfile);
    while not eof(gabfile)  do
    read(gabfile,garb);
    repeat
    clrscr;

writeln('Residents have to pay the following to be
registered:');
writeln('                    ↄ ');
writeln(' Initial connection fee - Ksh.500.00   ');
writeln(' Monthly subscription   - Ksh.100.00   ');
writeln('                    ↄ ');
clrscr;
writeln('Enter the ID number of the tenant who');
writeln('wants to be connected:');
read(id);
```

```
 writeln('The date of registration
is:',dd,'/',mm,'/',yyyy);
    garb.dtgreg.day:=dd;
    garb.dtgreg.month:=mm;
    garb.dtgreg.year:=yyyy;
    reset(regisfile);
    repeat
    read(regisfile,register);
    until register.resid=id;
     garb.name:=register.fname;
     garb.id:=register.resID;

     seek(gabfile,filesize(gabfile));
     seek(payfile,filesize(payfile));
     write(gabfile,garb);
     write(payfile,pay);
    writeln('DO YOU WANT TO REGISTER ANOTHER
RESIDENT?(Y/N)');
      repeat
        c:=upcase(readkey);
    until c in ['Y','N'];
     writeln(c);
    until c='N';
    close(gabfile);
    close(payfile);
      end;

Procedure registration;
var
ch:integer;
begin
clrscr;
   textcolor(yellow);
   gotoxy(25,3);writeln('REGISTRATION MENU');
   gotoxy(5,4);writeln('-----------------------------------
---------------------');
   textcolor(white);
  gotoxy(4,7);writeln('What do you want to register?');
   writeln('     1Tenant              ');
   writeln('     2Business            ');
   writeln('     3Sewer Services      ');
   writeln('     4Garbage collection  ');
   read(ch);
   if ch>4 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(ch);
```

```
        until ch<=4;
    case ch of
        1:input_res;
        2:input_bus;
        3:input_sewer;
        4:input_garbage;
     end;
     end;

procedure view_res;     {procedure for viewing tenant
details}
var x,y:integer;
begin
  dt;
  clrscr;textcolor(10);
        gotoxy(1,1);writeln('
ɔ ');
        gotoxy(1,2);write(' NAME
');
        gotoxy(15,2);write(' ID NO.
');
        gotoxy(27,2);write(' TEL NO.
');
        gotoxy(43,2);write(' HSE TYPE ');
        gotoxy(60,2);write(' DATE REGISTERED ');
        gotoxy(1,3);writeln('
ˌ' );
        reset(regisfile);
         x:=0;
        while not eof(regisfile) do
        begin
        read(regisfile,register);
        x:=x+1;
        textcolor(14);
        gotoxy(2,wherey);write(register.fname);
        gotoxy(16,wherey);write(register.resID:0:0);
        gotoxy(28,wherey);write(register.tel:0:0);
        gotoxy(44,wherey);write(register.hsetype);

gotoxy(61,wherey);writeln(register.dtreg.day,'/',register.d
treg.month,'/',register.dtreg.year);
        end;
        textcolor(10);
        y:=3;
        repeat
        y:=y+1;
        gotoxy(1,y);writeln('');
```

```
        gotoxy(15,y);writeln('');
        gotoxy(27,y);writeln('');
        gotoxy(43,y);writeln('');
        gotoxy(60,y);writeln('');
        gotoxy(78,y);writeln('');
        until y=x+3;
        gotoxy(1,wherey);writeln('
c');
        writeln('');writeln('');writeln('');
textcolor(white);
        Writeln('      Press any key to continue......');
        readkey;

end;

procedure view_regbus;    {procedure for viewing business
details}
var x,y:integer;
begin
  dt;
  clrscr;textcolor(10);
        gotoxy(1,1);writeln('
o ');
        gotoxy(1,2);write(' NAME');
        gotoxy(15,2);write('LIC NO.');
        gotoxy(27,2);write('BSN TYPE');
        gotoxy(47,2);write('TEL NO.');
        gotoxy(60,2);write('DATE REGISTERED');
         gotoxy(1,3);writeln('
' );
          reset(busfile);
          x:=0;
        while not eof(busfile) do
        begin
        read(busfile,business);
        x:=x+1;
        textcolor(14);
        gotoxy(2,wherey);write(business.name);
        gotoxy(16,wherey);write(business.busID:0:0);
        gotoxy(28,wherey);write(business.bstype);
        gotoxy(48,wherey);write(business.tel:0:0);

gotoxy(61,wherey);writeln(business.dtereg.day,'/',business.
dtereg.month,'/',business.dtereg.year);
        end;
        textcolor(10);
        y:=3;
```

```
        repeat
        y:=y+1;
        gotoxy(1,y);writeln('');
        gotoxy(15,y);writeln('');
        gotoxy(27,y);writeln('');
        gotoxy(47,y);writeln('');
        gotoxy(60,y);writeln('');
        gotoxy(76,y);writeln('');
        until y=x+3;
        gotoxy(1,wherey);writeln('
c');
        writeln('');writeln('');writeln('');
textcolor(white);
        Writeln('      Press any key to continue......');
        readkey;
        end;

  procedure view_regsew;    {procedure for viewing tenants
registered to sewage connection}
begin
  clrscr;textcolor(10);
        gotoxy(3,2);write('NAME');
        gotoxy(20,2);write('ID NO.');
        gotoxy(29,2);write('SUBSCRIPTION');
        gotoxy(52,2);writeln('DATE REGISTERED');
        reset(sewfile);
        while not eof(sewfile) do
        begin
        read(sewfile,sewage);
        begin
        textcolor(14);
        gotoxy(3,wherey);write(sewage.name);
        gotoxy(20,wherey);write(sewage.ID:0:0);
        gotoxy(29,wherey);write(sewage.subtype);

gotoxy(52,wherey);write(sewage.datreg.day,'/',sewage.datreg
.month,'/',sewage.datreg.year);
        end;
        end;
        writeln('');writeln('');writeln('');
textcolor(white);
        Writeln('      Press any key to continue......');
        readkey;
        end;

procedure view_regarb;   {procedure for viewing tenants
registered to garbage collection}
```

```
  begin
  clrscr;textcolor(10);
        gotoxy(3,2);write('NAME');
        gotoxy(20,2);write('ID NO.');
        gotoxy(29,2);write('DATE REGISTERED');
        reset(gabfile);
        while not eof(gabfile) do
        begin
        read(gabfile,garb);
        begin
        textcolor(14);
        gotoxy(3,wherey);write(garb.name);
        gotoxy(20,wherey);write(garb.ID:0:0);

gotoxy(29,wherey);writeln(garb.dtgreg.day,'/',garb.dtgreg.m
onth,'/',garb.dtgreg.year);
        end;
        end;
        writeln('');writeln('');writeln('');
textcolor(white);
        Writeln('      Press any key to continue......');
        readkey;
        end;

  Procedure display;
  var
    view:integer;
      begin
      clrscr;
        textcolor(yellow);
     gotoxy(25,3);writeln('DISPLAY RECORDS MENU');
     gotoxy(5,4);writeln('-------------------------------
-------------------------');
     textcolor(white);
        gotoxy(3,7);writeln('On which basis do you want to
view the records?');
        Writeln('  1.Registered tenants');
        writeln('  2.Registered businesses');
        writeln('  3.Residents registered to sewer
service');
        writeln('  4.Residents registered to gabage
collection');
        writeln;
        write('Choose from above:');readln(view);
        if view>4 then repeat
        writeln('ERROR!Enter valid choice');
        readln(view);until view<4;
```

```
    case view of
          1:view_res;
          2:view_regbus;
          3:view_regsew;
          4:view_regarb;
          end;
          end;


Procedure col_rent;   {procedure for collecting house rent
payments}
VAR
rnt,m,s:integer;
id:real;
begin
clrscr;

       reset(payfile);
       reset(regisfile);

    clrscr;
view_res;
writeln('Enter the ID number of the tenant');
readln(ID);
writeln('Enter the month for which payment is being made');
writeln('      1.January                  7.July');
writeln('      2.February                 8.August');
writeln('      3.March                    9.September');
writeln('      4.April                   10.October');
writeln('      5.May                     11.November');
writeln('      6.June                    12.December');

 readln(m);
 reset(regisfile);
 repeat
read(regisfile,register);
until register.resID=id;
writeln('The tenant is ',register.fname);
writeln('payment for ',month[m]);
repeat
read(payfile,pay);
until pay.resID=id; s:=0;
case m of
1:If pay.m1='paid' then s:=1;
2:If pay.m2='paid' then s:=1;
3:If pay.m3='paid' then s:=1;
4:If pay.m4='paid' then s:=1;
```

```
5:If pay.m5='paid' then s:=1;
6:If pay.m6='paid' then s:=1;
7:If pay.m7='paid' then s:=1;
8:If pay.m8='paid' then s:=1;
9:If pay.m9='paid' then s:=1;
10:If pay.m10='paid' then s:=1;
11:If pay.m11='paid' then s:=1;
12:If pay.m12='paid' then s:=1;
end;
If s=1 then
writeln('The tenant has already paid for ',month[m])  else
If s=0 then
Begin
    seek(payfile,filepos(payfile)-1);
    If register.hsetype='Single room' then pay.rent:=500
else
    if register.hsetype='One bedroomed' then pay.rent:=1500
else
    if register.hsetype='Two bedroomed' then pay.rent:=2500
else
    if register.hsetype='Three bedroomed' then
pay.rent:=3500;
    case m of
    1:pay.m1:='paid';
    2:pay.m2:='paid';
    3:pay.m3:='paid';
    4:pay.m4:='paid';
    5:pay.m5:='paid';
    6:pay.m6:='paid';
    7:pay.m7:='paid';
    8:pay.m8:='paid';
    9:pay.m9:='paid';
    10:pay.m10:='paid';
    11:pay.m11:='paid';
    12:pay.m12:='paid';
    end;
    dt;
    pay.dtpay.day:=dd;
    pay.dtpay.month:=mm;
    pay.dtpay.year:=yyyy;
    pay.monthpaid:=month[m];
    write(payfile,pay);
          close(payfile);
          close(regisfile);
writeln('The amount paid is Ksh.',pay.rent:0:2);
writeln('Date of payment: ',dd,'/',mm,'/',yyyy);
end;
```

```
readln;
end;


Procedure col_sewer;     {procedure for collecting sewer
payments}
var
m,s:integer;
id:real;
begin
dt;
    clrscr;
    view_res;
write('Enter the ID number of the tenant:');
readln(id);
reset(sewfile);
repeat
read(sewfile,sewage);
until sewage.ID=id;
writeln('The tenant is ',sewage.name);
If sewage.subtype='Annual subscription' then
begin
reset(payfile);
repeat
read(payfile,pay);
until pay.resID=id;
s:=0;
If pay.annual='paid' then s:=1;
If s=1 then
writeln('The tenant has already paid for ',yyyy)  else
If s=0 then
Begin
  pay.pfee:=3000;
  pay.annual:='paid';
    writeln('The amount paid is Ksh.',pay.pfee:0:2);
writeln('Date of payment: ',dd,'/',mm,'/',yyyy);
   end;
readkey;
end
else
if sewage.subtype='Monthly subscription' then
begin
writeln('Enter the month for which payment is being made');
writeln('      1.January                  7.July');
writeln('      2.February                 8.August');
writeln('      3.March                    9.September');
writeln('      4.April                   10.October');
```

```
writeln('      5.May                        11.November');
writeln('      6.June                       12.December');
readln(m);
reset(payfile);
repeat
read(payfile,pay);
until pay.resID=id; s:=0;
case m of
1:If pay.ms1='paid' then s:=1;
2:If pay.ms2='paid' then s:=1;
3:If pay.ms3='paid' then s:=1;
4:If pay.ms4='paid' then s:=1;
5:If pay.ms5='paid' then s:=1;
6:If pay.ms6='paid' then s:=1;
7:If pay.ms7='paid' then s:=1;
8:If pay.ms8='paid' then s:=1;
9:If pay.ms9='paid' then s:=1;
10:If pay.ms10='paid' then s:=1;
11:If pay.ms11='paid' then s:=1;
12:If pay.ms12='paid' then s:=1;
end;
If s=1 then
writeln('The tenant has already paid for ',month[m])  else
If s=0 then
Begin
  pay.pfee:=300;
 case m of
    1:pay.ms1:='paid';
    2:pay.ms2:='paid';
    3:pay.ms3:='paid';
    4:pay.ms4:='paid';
    5:pay.ms5:='paid';
    6:pay.ms6:='paid';
    7:pay.ms7:='paid';
    8:pay.ms8:='paid';
    9:pay.ms9:='paid';
    10:pay.ms10:='paid';
    11:pay.ms11:='paid';
    12:pay.ms12:='paid';
    end;
    writeln('The amount paid is Ksh.',pay.pfee:0:2);
writeln('Date of payment: ',dd,'/',mm,'/',yyyy);
   end;
     end;
        pay.cfee:=3500;
          dt;
          pay.dtpay.day:=dd;
```

```
          pay.dtpay.month:=mm;
          pay.dtpay.year:=yyyy;
          seek(payfile,filepos(payfile)-1);
          write(payfile,pay);
readln;
close(sewfile);
close(payfile);
end;

Procedure col_license;      {procedure for collecting
business license payments}
var
s,m:integer;
lic:real;
begin
dt;
clrscr;

       reset(payfile);
       reset(busfile);
    clrscr;

writeln('Enter the license number of the business');
readln(lic);

 reset(busfile);
repeat
read(busfile,business);
until business.busID=lic;
writeln('The business is: ',business.name);
repeat
read(payfile,pay);
until pay.busID=lic;
s:=0;
If pay.annualb='paid' then s:=1;
If s=1 then
writeln('The business has already paid for ',yyyy)  else
If s=0 then
Begin
  pay.pfee:=3000;
  pay.annualb:='paid';


    seek(payfile,filepos(payfile)-1);
If business.bstype='General shop' then pay.bsn:=3000 else
if business.bstype='Hardware shop' then pay.bsn:=4500 else
```

```
if business.bstype='Private institution' then
pay.bsn:=10000 else
if business.bstype='Miscellaneous' then pay.bsn:=2500;
writeln('The amount paid is Ksh.',pay.bsn:0:2);
writeln('Date of payment: ',dd,'/',mm,'/',yyyy);
end;

          dt;
          pay.dtpay.day:=dd;
          pay.dtpay.month:=mm;
          pay.dtpay.year:=yyyy;
          write(payfile,pay);
          close(payfile);
          close(busfile);

readln;
end;

Procedure col_markt;    {procedure for collecting market
payments}
begin
  clrscr;
  reset(mktfile);
  reset(payfile);
  while not eof(payfile)  do
    read(payfile,pay);
    clrscr;
    seek(payfile,filesize(payfile));
  writeln('How many vendors came to the market today?');
  readln(mkt.vnd);
  mkt.total:=50*mkt.vnd;
  pay.mkt:=mkt.total;
  writeln('The amount collected is Ksh.',mkt.total:4:2);
  readln;
  dt;
  pay.dtpay.day:=dd;
  pay.dtpay.month:=mm;
  pay.dtpay.month:=yyyy;
  write(mktfile,mkt);
  write(payfile,pay);
  close(mktfile);
  close(payfile);
end;

Procedure col_garb;    {procedure for collecting garbage
payments}
VAR
```

```
m,s:integer;
id:real;
begin
clrscr;
       reset(gabfile);
       reset(payfile);

writeln('Enter the ID number of the tenant');
readln(id);
writeln('Enter the month for which payment is being made');
writeln('      1.January                  7.July');
writeln('      2.February                 8.August');
writeln('      3.March                    9.September');
writeln('      4.April                   10.October');
writeln('      5.May                     11.November');
writeln('      6.June                    12.December');

 readln(m);
 reset(gabfile);
repeat
read(gabfile,garb);
until garb.ID=id;
writeln('The tenant is ',garb.name);
writeln('payment for ',month[m]);
repeat
read(payfile,pay);
until pay.resID=id; s:=0;
case m of
1:If pay.mg1='paid' then s:=1;
2:If pay.mg2='paid' then s:=1;
3:If pay.mg3='paid' then s:=1;
4:If pay.mg4='paid' then s:=1;
5:If pay.mg5='paid' then s:=1;
6:If pay.mg6='paid' then s:=1;
7:If pay.mg7='paid' then s:=1;
8:If pay.mg8='paid' then s:=1;
9:If pay.mg9='paid' then s:=1;
10:If pay.mg10='paid' then s:=1;
11:If pay.mg11='paid' then s:=1;
12:If pay.mg12='paid' then s:=1;
end;
If s=1 then
writeln('The tenant has already paid for ',month[m])  else
If s=0 then
Begin
    seek(payfile,filepos(payfile)-1);
    pay.gfee:=100;
```

```
 case m of
    1:pay.mg1:='paid';
    2:pay.mg2:='paid';
    3:pay.mg3:='paid';
    4:pay.mg4:='paid';
    5:pay.mg5:='paid';
    6:pay.mg6:='paid';
    7:pay.mg7:='paid';
    8:pay.mg8:='paid';
    9:pay.mg9:='paid';
    10:pay.mg10:='paid';
    11:pay.mg11:='paid';
    12:pay.mg12:='paid';
    end;
          pay.gregfee:=500;
          dt;
          pay.dtpay.day:=dd;
          pay.dtpay.month:=mm;
          pay.dtpay.year:=yyyy;
          write(payfile,pay);
writeln('The amount paid is Ksh.',pay.gfee:4:2);
writeln('Date of payment: ',dd,'/',mm,'/',yyyy);
end;
readln;
close(payfile);
close(gabfile);
end;

Procedure col_park;    {procedure for collecting parking
payments}
VAR
pak:integer;
begin
clrscr;
       reset(prkfile);
       reset(payfile);
    while not eof(payfile)  do
    read(payfile,pay);

    clrscr;
    seek(payfile,filesize(payfile));
writeln('Which type of vehicle is being parked ?');
writeln('       1:Car');
writeln('       2:Pickup');
writeln('       3:Lorry');
writeln('       4:Cart');
read(pak);
```

```
case pak of
          1:pay.park:=100;
          2:pay.park:=200;
          3:pay.park:=500;
          4:pay.park:=30;
          end;
writeln('The amount collected is Ksh.',pay.park:3:0);
readln;
readln;
              dt;
          pay.dtpay.day:=dd;
          pay.dtpay.month:=mm;
          pay.dtpay.year:=yyyy;
          write(payfile,pay);
          close(prkfile);
          close(payfile);
end;

Procedure collection;
VAR
serv:integer;
begin
clrscr;
textcolor(yellow);
gotoxy(25,3);writeln('REVENUE COLLECTION MENU');
gotoxy(5,4);writeln('-------------------------------------
---------------------');
textcolor(white);
writeln('For which services do you want to collect revenue
?');
    writeln('           ɔ');
    writeln(' 1Rent              ');
    writeln(' 2Sewer             ');
    writeln(' 3Garbage           ');
    writeln(' 4License           ');
    writeln(' 5Market            ');
    writeln(' 6Parking           ');
    writeln('           ɔ');
    read(serv);
    if serv>6 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(serv);
       until serv<=6;
   case serv of
          1:col_rent;
          2:col_sewer;
```

```
          3:col_garb;
          4:col_license;
          5:col_markt;
          6:col_park;
          end;
end;

Procedure rep_daily;     {procedure for viewing daily
reports}
Var
rtsum,bussum,gbsum,mktsum,sewsum,paksum,total:real;
d,m,y:integer;
month,s:string;
begin
dt;
rtsum:=0; bussum:=0; gbsum:=0; mktsum:=0; sewsum:=0;
paksum:=0;
total:=0;
clrscr;
  gotoxy(2,4);textcolor(white);write('Enter the month : ');
      gotoxy(2,5);write('1.January ');
gotoxy(30,5);write('7. July ');
      gotoxy(2,7);write('2. February ');
gotoxy(30,7);write('8. August ');
      gotoxy(2,9);write('3. March ');
gotoxy(30,9);write('9. September ');
      gotoxy(2,11);write('4. April ');
gotoxy(30,11);write('10. October ');
      gotoxy(2,13);write('5. May');
gotoxy(30,13);writeln('11. November ');
      gotoxy(2,15);writeln('6. June ');
gotoxy(30,15);write('12. December ');
     writeln;
     writeln;
     writeln('Choice>>> ');
     readln(m);
 case m of

1:month:='January';2:month:='February';3:month:='March';4:m
onth:='April';5:month:='May';

6:month:='June';7:month:='July';8:month:='August';9:month:=
'September';10:month:='October';
      11:month:='November';12:month:='December';
end;{end case}
     write('Enter the year: ');
     readln(y);
```

```
     write('Enter the day: ');
     readln(d);

clrscr;
       gotoxy(2,3);write('Payment made on:'
,d,'/',m,'/',y);
       textcolor(yellow);
       gotoxy(10,7);write('Rent ');
       gotoxy(22,7);write('Business ');
       gotoxy(34,7);write('Garbage ');
       gotoxy(46,7);write('Sewer');
       gotoxy(57,7);write('Market');
       gotoxy(68,7);writeln('Parking ');
       s:='paid';
     reset(payfile);
while not eof(payfile) do
begin
   read(payfile,pay);
   if (pay.dtpay.month=m) and (pay.dtpay.year=y)
and(pay.dtpay.day=d) then
   begin
   if pay.m1=s then rtsum:=rtsum+pay.rent;
   if pay.m2=s then rtsum:=rtsum+pay.rent;
   if pay.m3=s then rtsum:=rtsum+pay.rent;
   if pay.m4=s then rtsum:=rtsum+pay.rent;
   if pay.m5=s then rtsum:=rtsum+pay.rent;
   if pay.m6=s then rtsum:=rtsum+pay.rent;
   if pay.m7=s then rtsum:=rtsum+pay.rent;
   if pay.m8=s then rtsum:=rtsum+pay.rent;
   if pay.m9=s then rtsum:=rtsum+pay.rent;
   if pay.m10=s then rtsum:=rtsum+pay.rent;
   if pay.m11=s then rtsum:=rtsum+pay.rent;
   if pay.m12=s then rtsum:=rtsum+pay.rent;
   bussum:=bussum+pay.bsn;
   if pay.mg1=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg2=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg3=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg4=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg5=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg6=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg7=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg8=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg9=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg10=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg11=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg12=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
```

```
   if pay.ms1=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms2=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms3=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms4=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms5=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms6=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms7=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms8=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms9=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms10=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms11=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms12=s then sewsum:=sewsum+pay.pfee+pay.cfee;
       mktsum:=mktsum+pay.mkt;
       paksum:=paksum+pay.park;
   end;
end;
        total:=rtsum+bussum+sewsum+gbsum+mktsum+paksum ;
        textcolor(lightgreen);
        gotoxy(10,8);write('Ksh.',rtsum:2:0);
        gotoxy(22,8);write('Ksh.',bussum:2:0);
        gotoxy(34,8);write('Ksh.',gbsum:2:0);
        gotoxy(46,8);write('Ksh.',sewsum:2:0);
        gotoxy(57,8);write('Ksh.',mktsum:2:0);
        gotoxy(68,8);writeln('Ksh.',paksum:2:0);
     writeln;
     textcolor(white);
     gotoxy(25,15);writeln('TOTAL REVENUE COLLECTED.
Ksh.',total:2:0);
     readln;
close(payfile);
 end;

Procedure rep_monthly; {procedure for viewing monthly
reports}
  var
   rtsum,gbsum,sewsum,total:real;
   m,y:integer;
   s:string;
 begin
 rtsum:=0; gbsum:=0;
 sewsum:=0;total:=0;
  clrscr;
  gotoxy(2,3);textcolor(white);writeln('Enter the Month :
');

      gotoxy(2,5);write('1.January ');
gotoxy(30,5);write('7. July ');
```

```
      gotoxy(2,7);write('2. February ');
gotoxy(30,7);write('8. August ');
      gotoxy(2,9);write('3. March ');
gotoxy(30,9);write('9. September ');
      gotoxy(2,11);write('4. April ');
gotoxy(30,11);write('10. October ');
      gotoxy(2,13);write('5. May');
gotoxy(30,13);writeln('11. November ');
      gotoxy(2,15);writeln('6. June ');
gotoxy(30,15);write('12. December ');
     writeln; writeln;
     write('Choice>>> : ');
     readln(m);

  write('Enter the year:' );
     readln(y);
begin
clrscr;
textcolor(white);
 gotoxy(2,3);write('Payments made in the Month of: ',
month[m], '   Year:', y);
 textcolor(yellow);
  gotoxy(10,7);write('Rent');
  gotoxy(22,7);write('Garbage');
  gotoxy(34,7);write('Sewer');
  s:='paid';
  reset(payfile);
while not eof(payfile) do
begin
    read(payfile,pay);
    if (pay.monthpaid=month[m]) and (pay.dtpay.year=y) then
     begin
     case m of
   1:if pay.m1=s then rtsum:=rtsum+pay.rent;
   2: if pay.m2=s then rtsum:=rtsum+pay.rent;
   3: if pay.m3=s then rtsum:=rtsum+pay.rent;
   4: if pay.m4=s then rtsum:=rtsum+pay.rent;
   5: if pay.m5=s then rtsum:=rtsum+pay.rent;
   6: if pay.m6=s then rtsum:=rtsum+pay.rent;
   7: if pay.m7=s then rtsum:=rtsum+pay.rent;
   8: if pay.m8=s then rtsum:=rtsum+pay.rent;
   9: if pay.m9=s then rtsum:=rtsum+pay.rent;
   10:if pay.m10=s then rtsum:=rtsum+pay.rent;
   11:if pay.m11=s then rtsum:=rtsum+pay.rent;
   12:if pay.m12=s then rtsum:=rtsum+pay.rent;
   end;
```

```
   if pay.mg1=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg2=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg3=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg4=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg5=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg6=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg7=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg8=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg9=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg10=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg11=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg12=s then gbsum:=gbsum+pay.gfee+pay.gregfee;

   if pay.ms1=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms2=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms3=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms4=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms5=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms6=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms7=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms8=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms9=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms10=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms11=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms12=s then sewsum:=sewsum+pay.pfee+pay.cfee;

   end;
   end;
        total:=rtsum+sewsum+gbsum;
        textcolor(lightgreen);
        gotoxy(10,8);write('Ksh.',rtsum:2:0);
        gotoxy(22,8);write('Ksh.',gbsum:2:0);
        gotoxy(34,8);write('Ksh.',sewsum:2:0);
     writeln;
     textcolor(white);
     gotoxy(20,15);writeln('TOTAL REVENUE COLLECTED:
Ksh.',total:0:0);
     readln;
 close(payfile);
 end;
 end;

Procedure rep_annual; {procedure for viewing annual
reports}
  var
   rtsum,bussum,mktsum,paksum,gbsum,sewsum,total:real;
   m,y:integer;
```

```
   s:string;
 begin
 rtsum:=0; bussum:=0;mktsum:=0; paksum:=0; gbsum:=0;
 sewsum:=0;total:=0;
  clrscr;
  writeln('Enter the year:' );
     textcolor(white);readln(y);
begin
clrscr;
textcolor(white); gotoxy(2,3);write('Payments made in the
Year:', y);
  gotoxy(2,5);write('SERVICES');
  gotoxy(25,5);write('AMOUNT COLLECTED');
  gotoxy(2,6);write('Rent');
  gotoxy(2,7);write('Business');
  gotoxy(2,8);write('Garbage');
  gotoxy(2,9);write('Sewer');
  gotoxy(2,10);write('Market');
  gotoxy(2,11);write('Parking');
textcolor(yellow);

     reset(payfile);
begin
s:='paid';
while not eof(payfile) do
begin
read(payfile,pay);

   if pay.m1=s then rtsum:=rtsum+pay.rent;
   if pay.m2=s then rtsum:=rtsum+pay.rent;
   if pay.m3=s then rtsum:=rtsum+pay.rent;
   if pay.m4=s then rtsum:=rtsum+pay.rent;
   if pay.m5=s then rtsum:=rtsum+pay.rent;
   if pay.m6=s then rtsum:=rtsum+pay.rent;
   if pay.m7=s then rtsum:=rtsum+pay.rent;
   if pay.m8=s then rtsum:=rtsum+pay.rent;
   if pay.m9=s then rtsum:=rtsum+pay.rent;
   if pay.m10=s then rtsum:=rtsum+pay.rent;
   if pay.m11=s then rtsum:=rtsum+pay.rent;
   if pay.m12=s then rtsum:=rtsum+pay.rent;
     bussum:=bussum+pay.bsn;
   if pay.mg1=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg2=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg3=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg4=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg5=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg6=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
```

```
   if pay.mg7=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg8=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg9=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg10=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg11=s then gbsum:=gbsum+pay.gfee+pay.gregfee;
   if pay.mg12=s then gbsum:=gbsum+pay.gfee+pay.gregfee;

   if pay.ms1=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms2=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms3=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms4=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms5=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms6=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms7=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms8=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms9=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms10=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms11=s then sewsum:=sewsum+pay.pfee+pay.cfee;
   if pay.ms12=s then sewsum:=sewsum+pay.pfee+pay.cfee;
       mktsum:=mktsum+pay.mkt;
       paksum:=paksum+pay.park;
   end;
end;

        total:=rtsum+bussum+sewsum+gbsum+mktsum+paksum;
        gotoxy(25,6);write('Ksh.',rtsum:2:0);
        gotoxy(25,7);write('Ksh.',bussum:2:0);
        gotoxy(25,8);write('Ksh.',gbsum:2:0);
        gotoxy(25,9);write('Ksh.',sewsum:2:0);
        gotoxy(25,10);write('Ksh.',mktsum:2:0);
        gotoxy(25,11);writeln('Ksh.',paksum:2:0);
        writeln;
  gotoxy(25,15);writeln('TOTAL REVENUE COLLECTED.
Ksh.',total:0:0);
  readln;
  close(payfile);
 end;
 end;

Procedure reports;
var
r:integer;
begin
  clrscr;
  textcolor(yellow);
 gotoxy(29,3);writeln('REPORTS MENU');
```

```
 gotoxy(5,4);writeln('-----------------------------------
-----------------------------');
  writeln;
  writeln;
  textcolor(white);
 gotoxy(5,7);writeln('  Which reports do you want to view?
');
  writeln('      1.Daily reports     ');
  writeln('      2.Monthly reports   ');
  writeln('      3.Annual reports     ');
  readln (r);
  if r>3 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(r);
       until r<=3;
    case r of
    1:rep_daily;
    2:rep_monthly;
    3:rep_annual;
    end;
end;

Procedure edit_ten;  {procedure for editing tenant details}
 var
 hs,ndai,counter:integer;
 ind:real;
 begin
       reset(regisfile);
      clrscr;
      view_res;
      Writeln('Enter the ID no. of the business you want to
edit');
      write('ID NO.:');readln(ind);
      repeat
      counter:=0; reset(regisfile);
      repeat
      read(regisfile,register);
      if register.resID=ind then
      counter:=1;
      until eof(regisfile);
      if counter=0 then
      Begin
      writeln('ERROR! INVALID ENTRY');
      writeln('Enter an ID number that
exists');readln(ind);end;
      until counter=1;
```

```
      reset(regisfile);
      repeat
      read(regisfile,register);
      counter:=counter+1;
      until register.resID=ind;


      begin

      seek(regisfile,filepos(regisfile)-1);
      writeln('       ****Re-enter the following tenant
details to edit****');
      clrscr;
         writeln('Enter the name of the resident');
    readln(register.fname);
    writeln('Enter the telephone number of the resident');
    readln(register.tel);
    writeln('Enter the type of house');
    writeln('           HOUSES:                      ');
    writeln('                    ↄ');
    writeln('    House Type            Rent        ');
    writeln(' 1Single room      Ksh.500.00     ');
    writeln(' 2One bedroomed    Ksh.1,500.00   ');
    writeln(' 3Two bedroomed    Ksh.2,500.00   ');
    writeln(' 4Three bedroomed  Ksh.3,500.00   ');
    writeln('                    ↄ');
    write('House type:');read(hs);
    case hs of
          1:register.hsetype:='Single room';
          2:register.hsetype:='One bedroomed';
          3:register.hsetype:='Two bedroomed';
          4:register.hsetype:='Three bedroomed';
          end;

      write(regisfile,register);
      close(regisfile);
      end
      end;

Procedure edit_bus;    {procedure for editing business
details}
var
 bs,gari,counter:integer;
 lic:real;
 begin
```

```
      reset(busfile);
      clrscr;
      view_regbus;

      Writeln('Enter the license no. of the business you
want to edit');
      write('LIC NO.:');readln(lic);
      repeat
      counter:=0; reset(busfile);
      repeat
      read(busfile,business);
      if business.busID=lic then
      counter:=1;
      until eof(busfile);
      if counter=0 then
      Begin
      writeln('ERROR! INVALID ENTRY');
      writeln('Enter a license number that
exists');readln(lic);end;
      until counter=1;
      reset(busfile);
      repeat
      read(busfile,business);
      counter:=counter+1;
      until business.busID=lic;


      begin
      seek(busfile,filepos(busfile)-1);
      writeln('       ****Re-enter the following tenant
details to edit****');
      clrscr;
       writeln('Enter the name of the business');
    readln(business.name);
    writeln('Enter the license number of the business');
    readln(business.busID);
    writeln('Enter the type of business preffered');
     writeln('       BUSINESSES:                    ');
    writeln('                  ɔ');
    writeln('  Business type       License fee     ');
    writeln(' 1:General shops         Ksh.500.00   ');
    writeln(' 2:Hardware shops        Ksh.1,500.00 ');
    writeln(' 3:Private institutions  Ksh.2,500.00 ');
    writeln(' 4:Miscellaneous         Ksh.3,500.00 ');
    writeln('                  ɔ');
    write('Business type:');read(bs);
     case bs of
```

```
          1:business.bstype:='General shops';
          2:business.bstype:='Hardware shops';
          3:business.bstype:='Private institutions';
          4:business.bstype:='Miscellaneous';
          end;

    writeln('Enter the telephone no. of the business');
    readln(business.tel);
      write(busfile,business);
      close(busfile);
      end;
      end;

Procedure edit;
var
red:integer;
begin
     clrscr;
     textcolor(yellow);
     gotoxy(29,3);writeln('EDITING MENU');
     gotoxy(5,4);writeln('--------------------------------
--------------------------------');
     writeln;
     writeln;
     textcolor(white);
     gotoxy(4,7);writeln('Choose the criteria you want to
edit');
     writeln('        1.Tenant records');
     writeln('        2.Business records');
     read(red);
     if red>2 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(red);
       until red<=2;
     case red of
     1:edit_ten;
     2:edit_bus;
     end;
 end;

Procedure del_ten;    {procedure for deleting tenant
records}
 var
  n,r:integer;
  id:real;
begin
```

```
      clrscr;
     view_res;
      Writeln('Enter the ID number of the tenant you want
to delete ');
      write('ID NO.:');readln(id);
      repeat
      n:=0; reset(regisfile);
      repeat
      read(regisfile,register);
      if register.resID=id then
      n:=1;
      until eof(regisfile);
      if n=0 then
      Begin
      writeln('ERROR! INVALID ENTRY');
      writeln('Enter an ID number that
exists');readln(id);end;
      until n=1;
      reset(regisfile);
      repeat
      read(regisfile,register);
      n:=n+1;
      until register.resID=id;

      for r:=n to filesize(regisfile)-1 do
      begin
      seek(regisfile,r);
      read(regisfile,register);
      seek(regisfile,n-1);
      write(regisfile,register);
      end;
      seek(regisfile,filesize(regisfile)-1);
      truncate(regisfile);
      writeln('The resident records have been deleted');
      readln;
      end;


Procedure del_bus;     {procedure for deleting tenant
records}
var
  n,r:integer;
  id:real;
begin
```

```
     clrscr;
     view_regbus;

      Writeln('Enter the license number of the business you
want to delete ');
      write('LICENSE NO.:');readln(id);
       repeat
      n:=0; reset(busfile);
      repeat
      read(busfile,business);
      if business.busID=id then
      n:=1;
      until eof(busfile);
      if n=0 then
      Begin
      writeln('ERROR! INVALID ENTRY');
      writeln('Enter a license number that
exists');readln(id);end;
      until n=1;
      reset(busfile);
      repeat
      read(busfile,business);
      n:=n+1;
      until business.busID=id;

      for r:=n to filesize(busfile)-1 do
      begin
      seek(busfile,r);
      read(busfile,business);
      seek(busfile,n-1);
      write(busfile,business);
      end;
      seek(busfile,filesize(busfile)-1);
      truncate(busfile);
      end;
Procedure delete;
VAR
rec:integer;
begin
clrscr;
     textcolor(yellow);
     gotoxy(29,3);writeln('DELETE RECORDS MENU');
     gotoxy(5,4);writeln('---------------------------------
---------------------------------');
     writeln;
     writeln;
     textcolor(white);
```

```
     gotoxy(4,7);writeln('Which records do you want to
delete ?');
     writeln('        1.Tenant records');
     writeln('        2.Business records');
     read(rec);
     if rec>2 then
       repeat
       writeln('ERROR! PLEASE ENTER THE CORRECT TASK');
       readln(rec);
       until rec<=2;
     case rec of
     1:del_ten;
     2:del_bus;
     end;
 end;
Procedure exit;     {procedure for exiting program}
var
  Gd, Gm, I, Width: Integer;
begin
  Gd := Detect; InitGraph(Gd, Gm, '');
  if GraphResult <> grOk then Halt(1);
    setbkcolor(blue);
    setfillstyle(10,blue);
    Bar(0, 0, 650, 600);
    setcolor(yellow);
    settextstyle(7,0,4);
    outtextxy(150,200,'HAVE A NICE DAY');
  ReadLn;
  CloseGraph;
end;


Procedure assigning;       {procedure for assigning files}
begin
assign(busfile,'bus.dat');
assign(regisfile,'regis.dat');
assign(payfile,'pay.dat');
assign(sewfile,'sew.dat');
assign(gabfile,'gab.dat');
assign(mktfile,'mkt.dat');
assign(prkfile,'prk.dat');
end;
BEGIN
begin
entry
end;
```

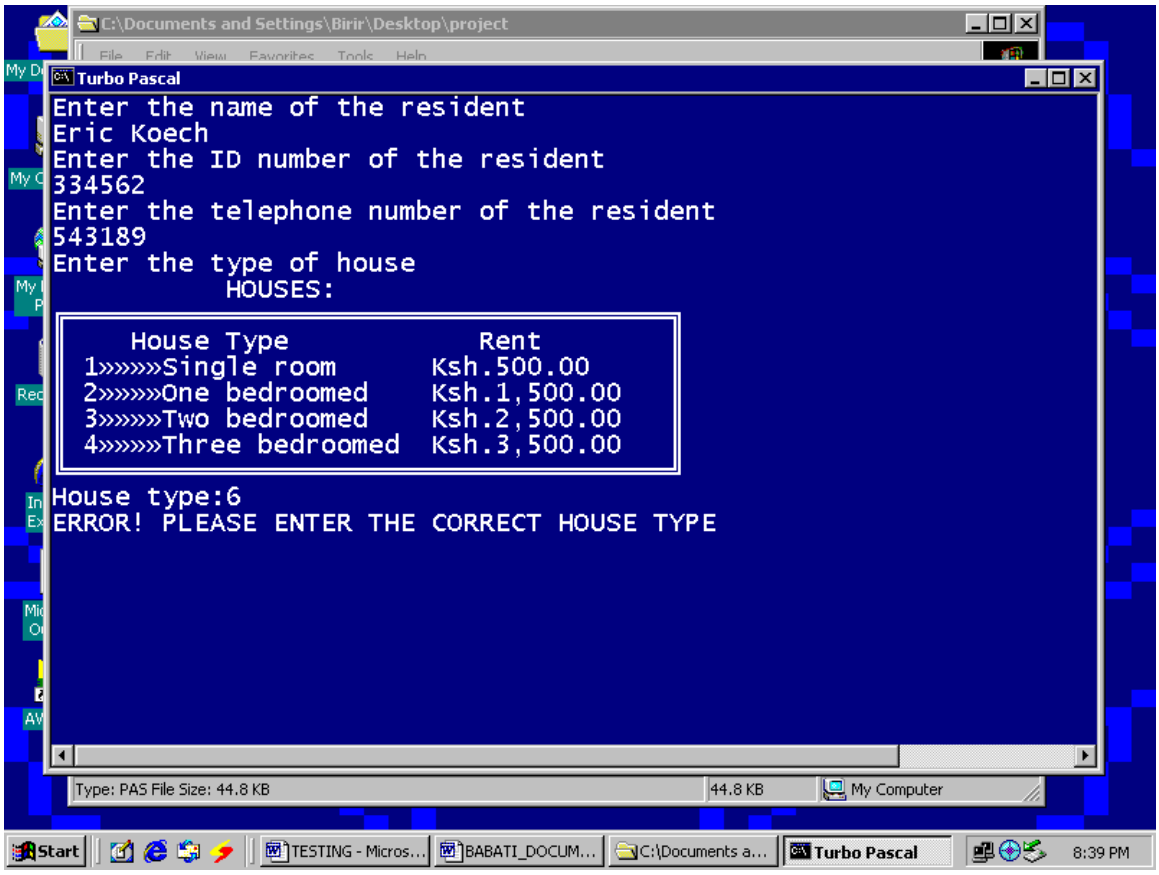
```
clrscr;
assigning;
menu;
repeat
case tsk of
1:Begin
  registration;
  menu;
  end;
2:Begin
  collection;
  menu;
  end;
3:Begin
  display;
  menu;
  end;
4:Begin
  reports;
  menu;
  end;
5:Begin
  edit;
  menu;
  end;
6:Begin
  delete;
  menu;
  end;
7:Begin
  exit;
  end;
  end;
until tsk=7;

END.
```

6.TESTING

To make sure that Babati Urban Council revenue collection system is working as expected, it has to be tested.
During testing the following steps are taken:

(a) Registration testing

I) Incorrect tenant registration:

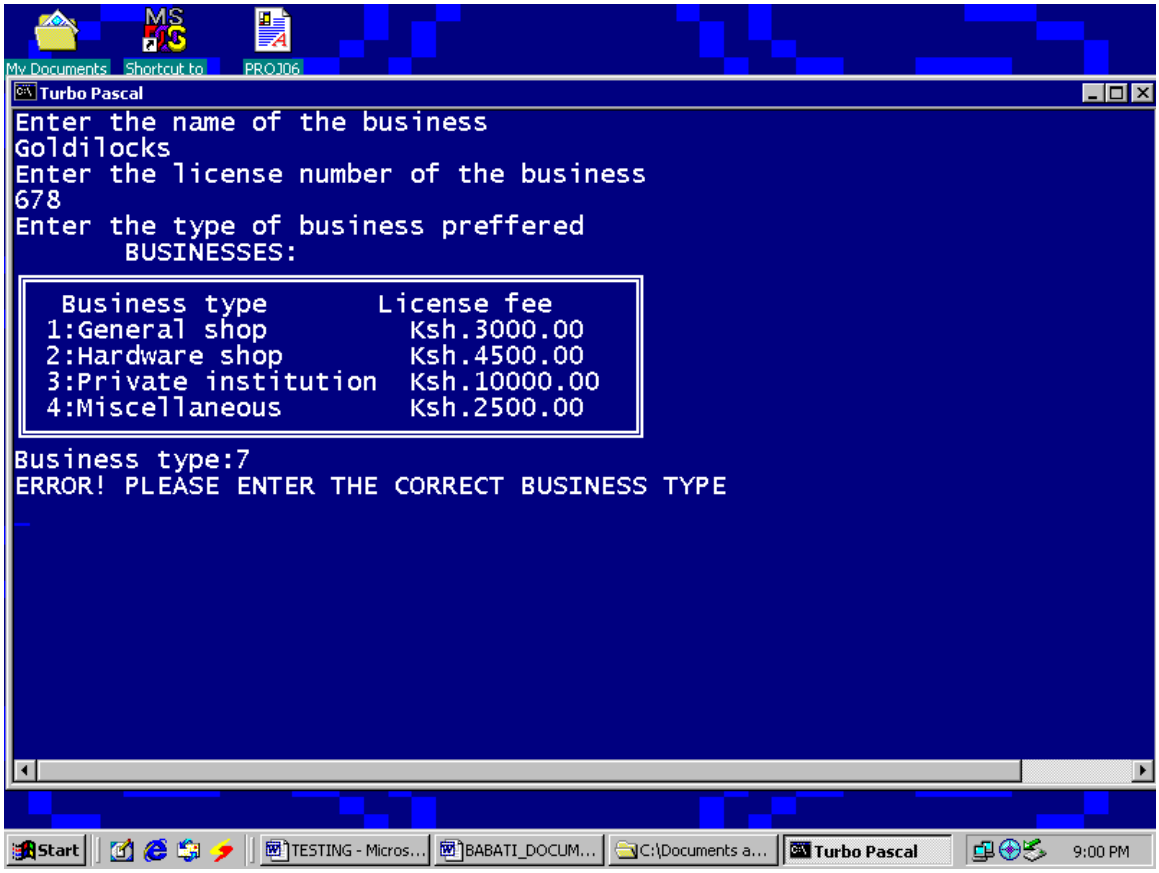


Invalid entry of houses since all choices should be between 1 & 4

II) Correct tenant registration:

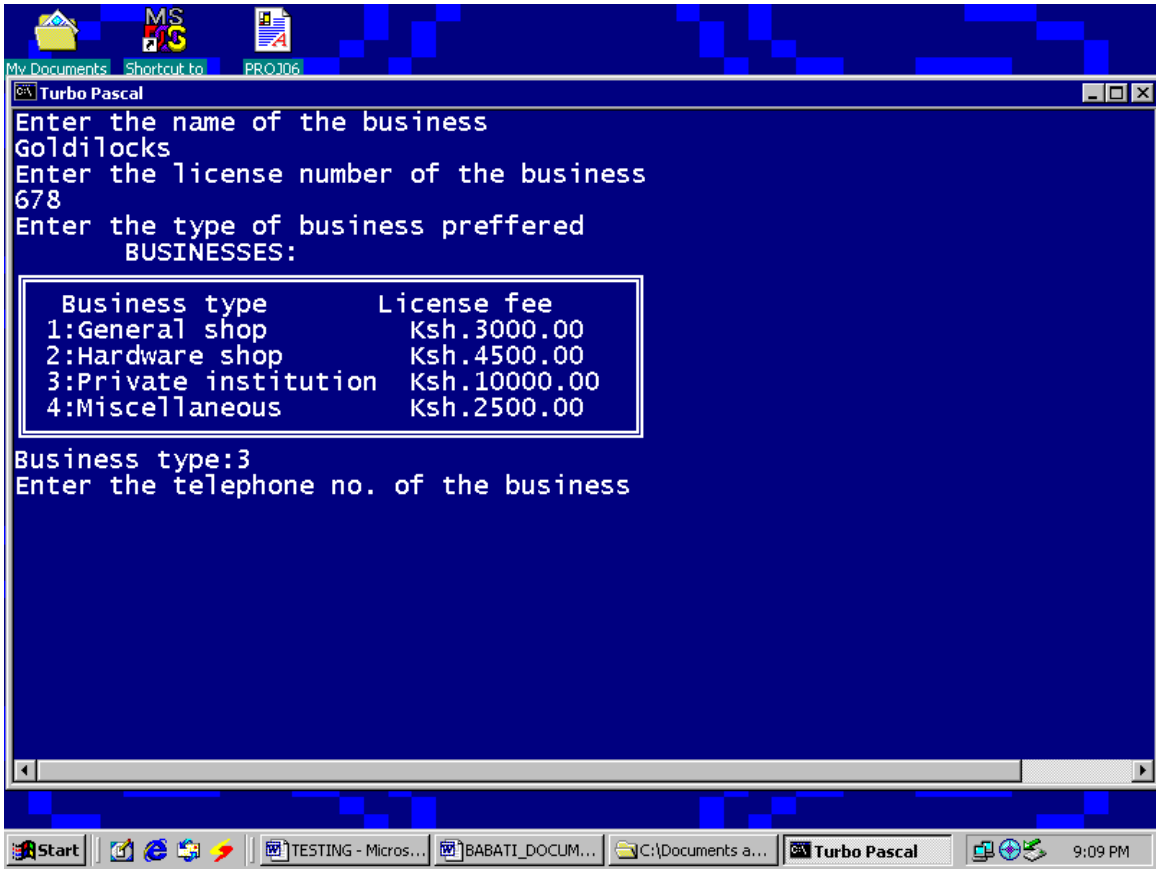
The program has accepted to register since house type is within the range

III) Incorrect business licensing registration:



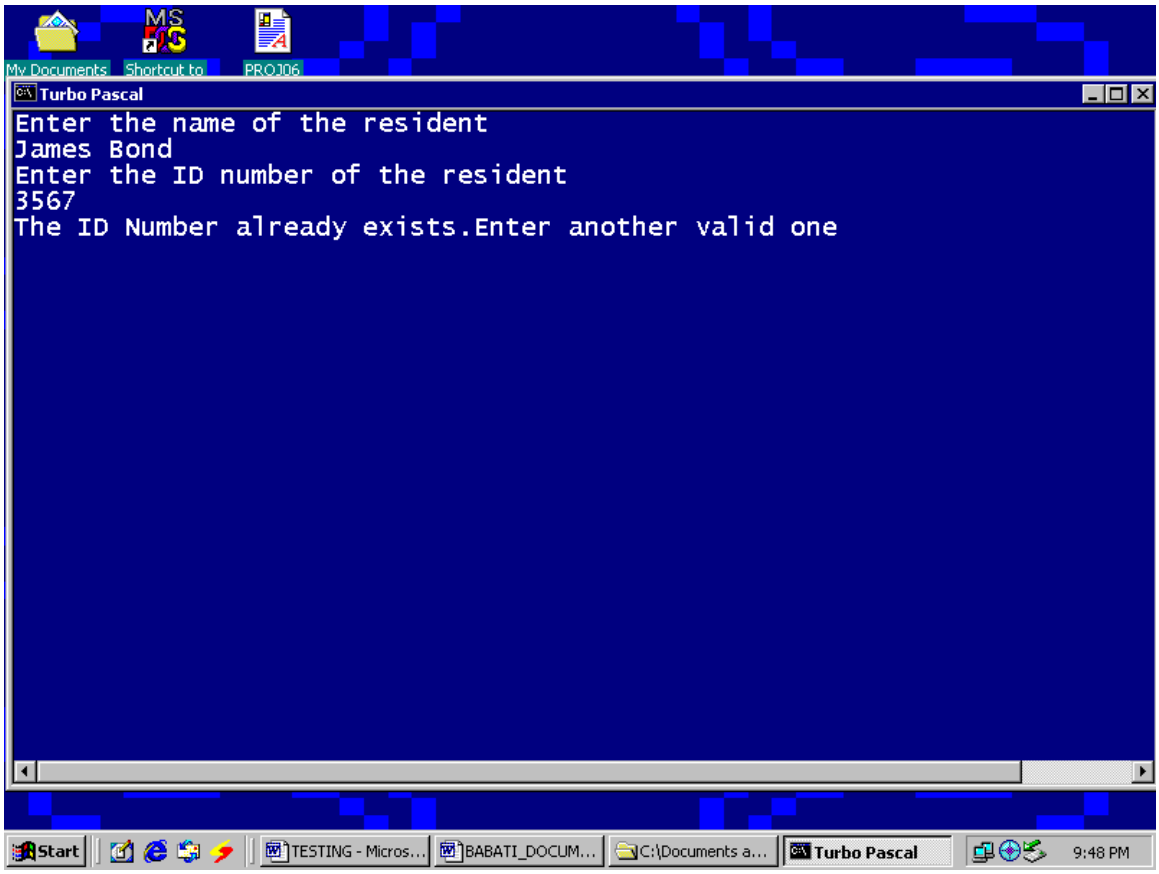
Invalid entry of business type since the choices are between 1 & 4

IV) Correct business licensing registration:



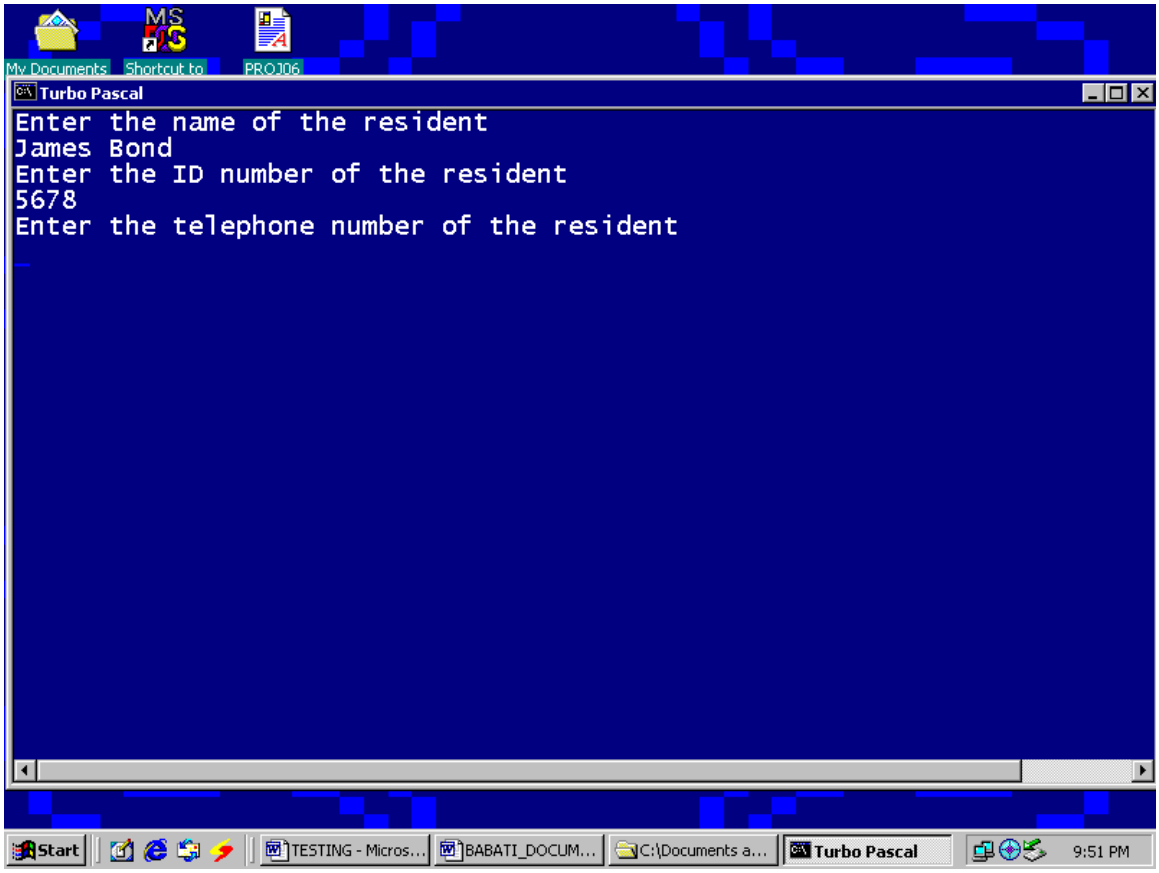
Program has accepted to register since choice is within range of business type

V) Invalid ID number:



If an ID number already exists the program won't register

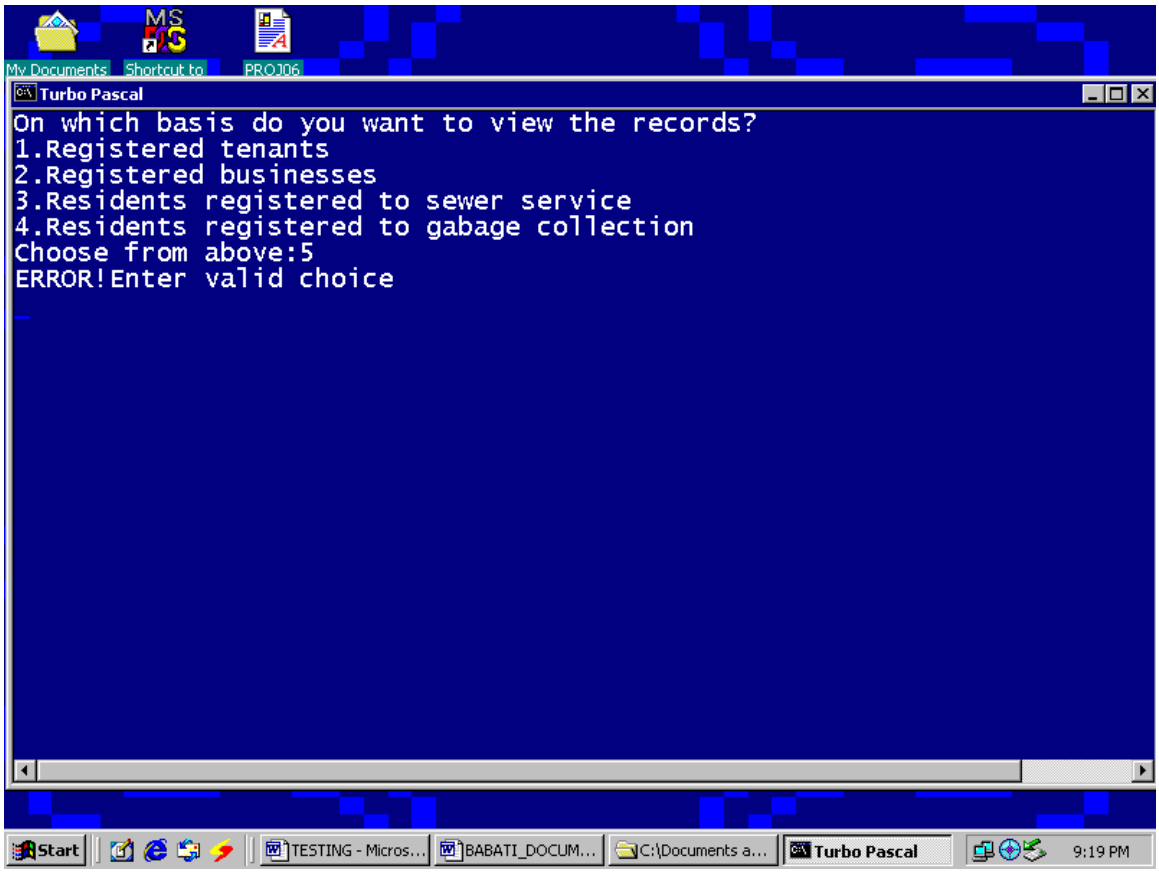
VI) Valid ID number:



If the ID number entered is valid, registration process will continue

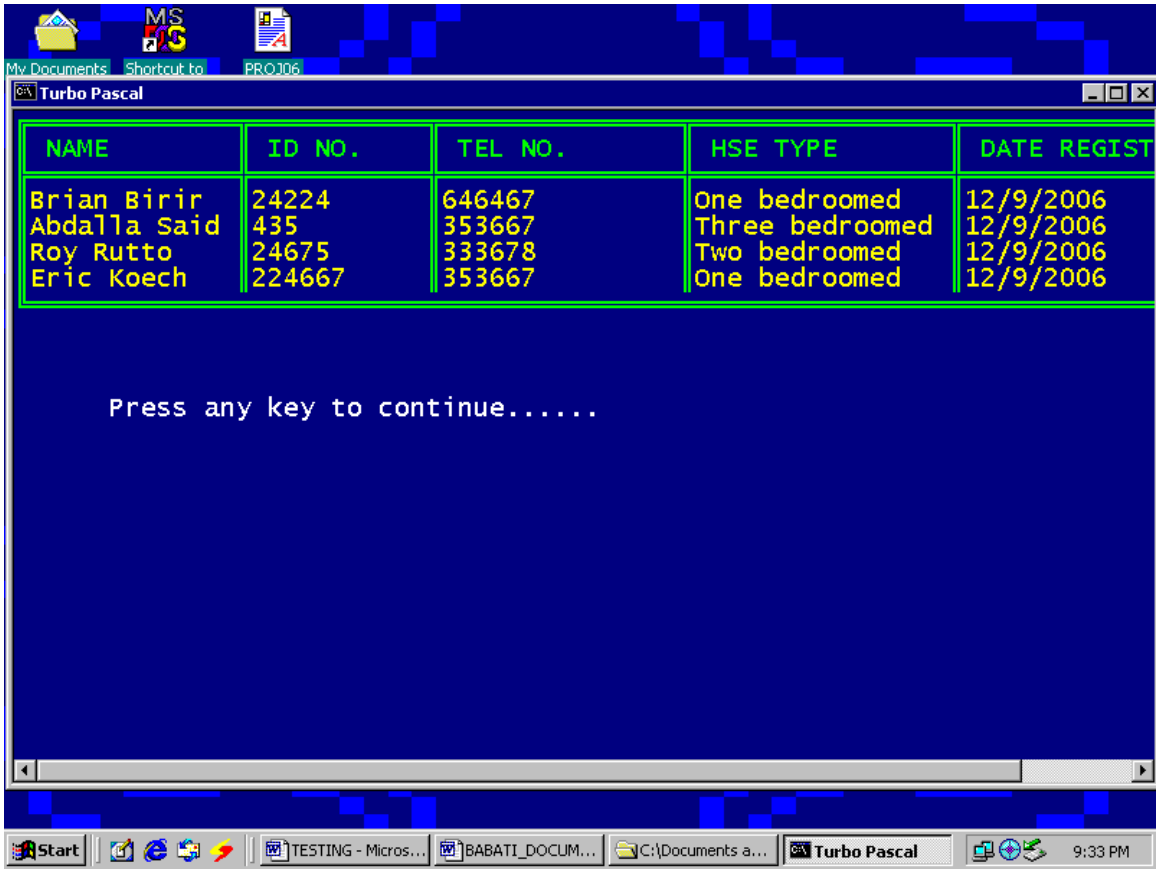
(b) Display testing

I) Incorrect choice of display:



Invalid input of choice since range is between 1 & 4

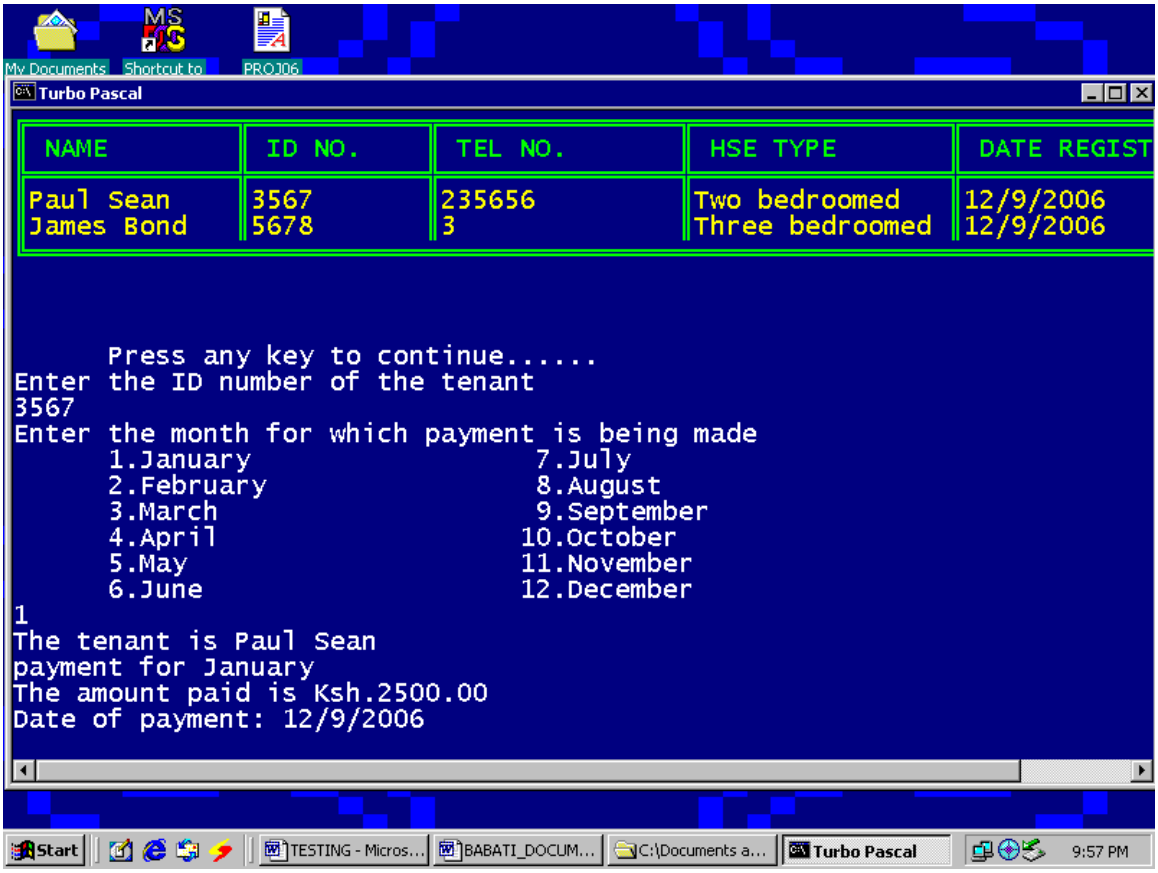
II) Correct choice of display:



The correct choice made will display details of files

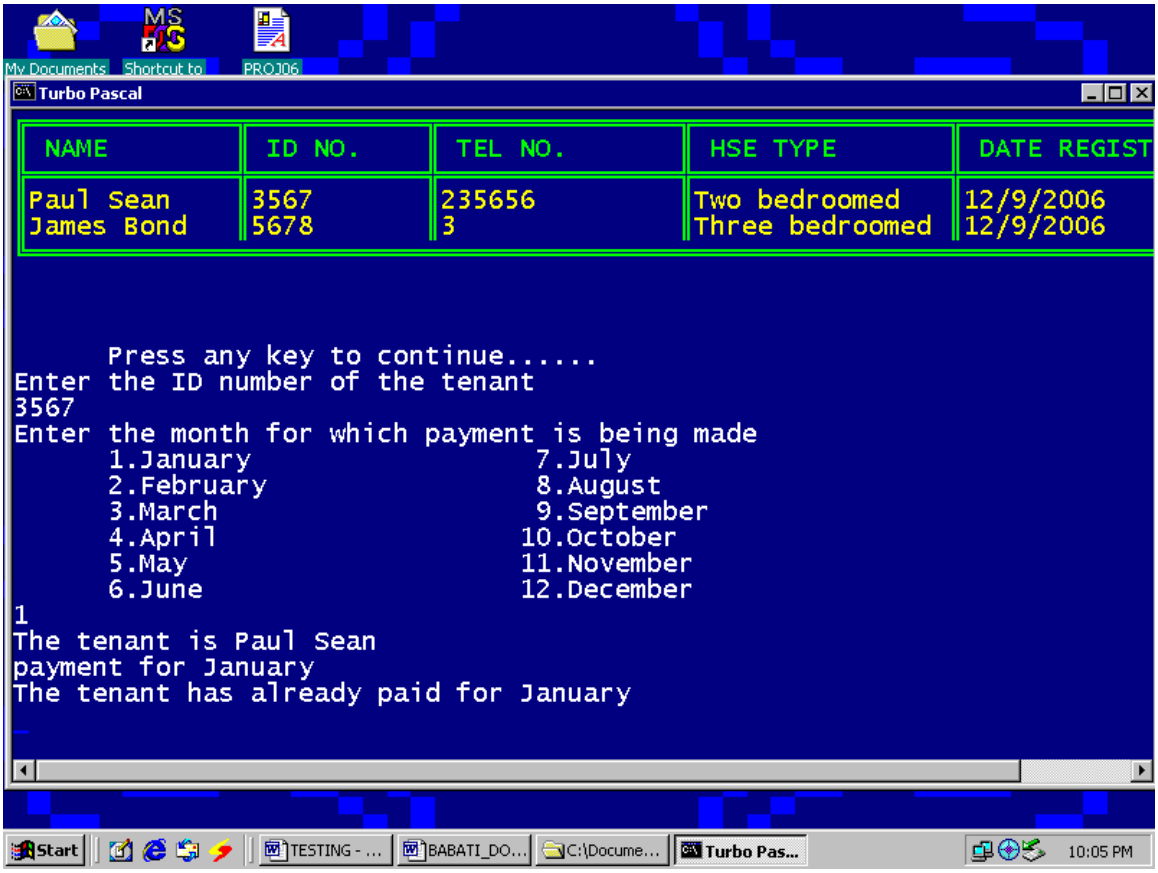
(c) Collecting revenue

I) Collecting revenue for first time:



The program accepts payment since it is collecting for the first time

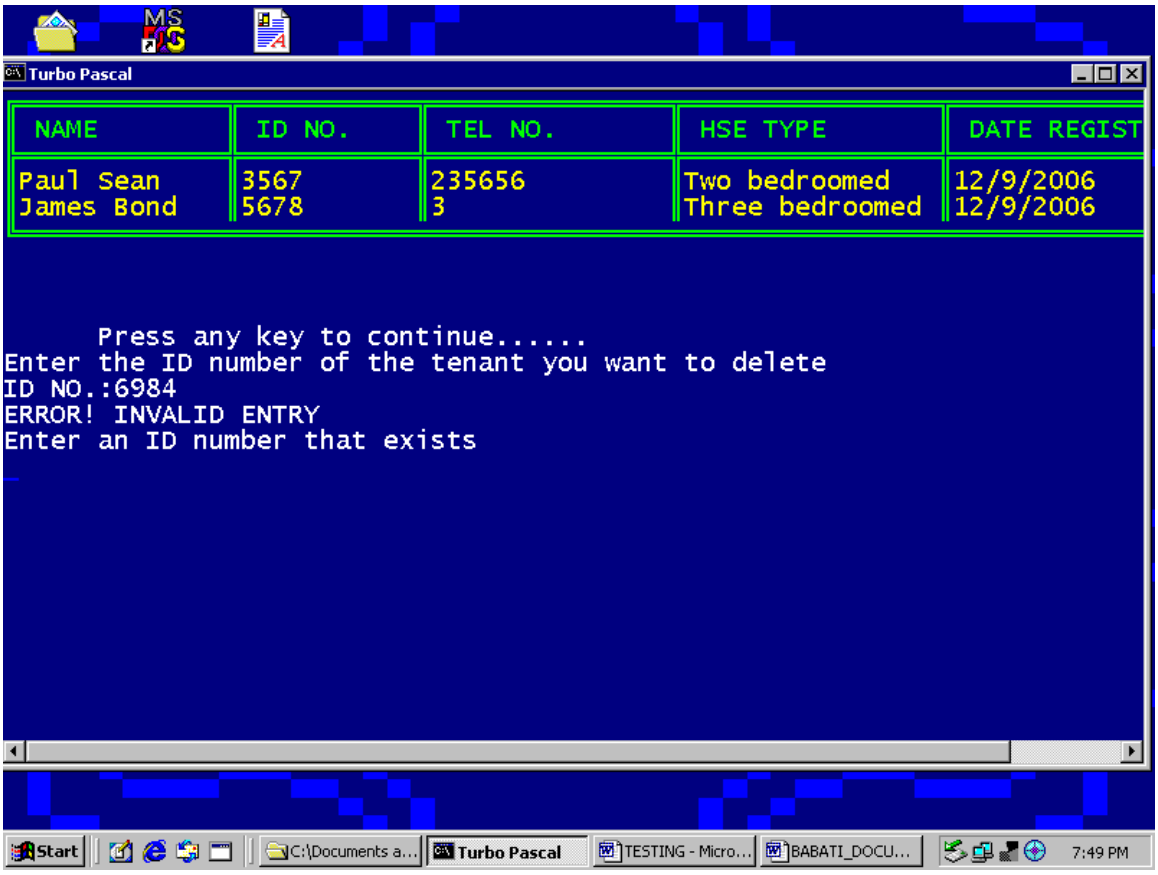
II) Collecting revenue if already paid:



The program will not accept payment since it has already been paid

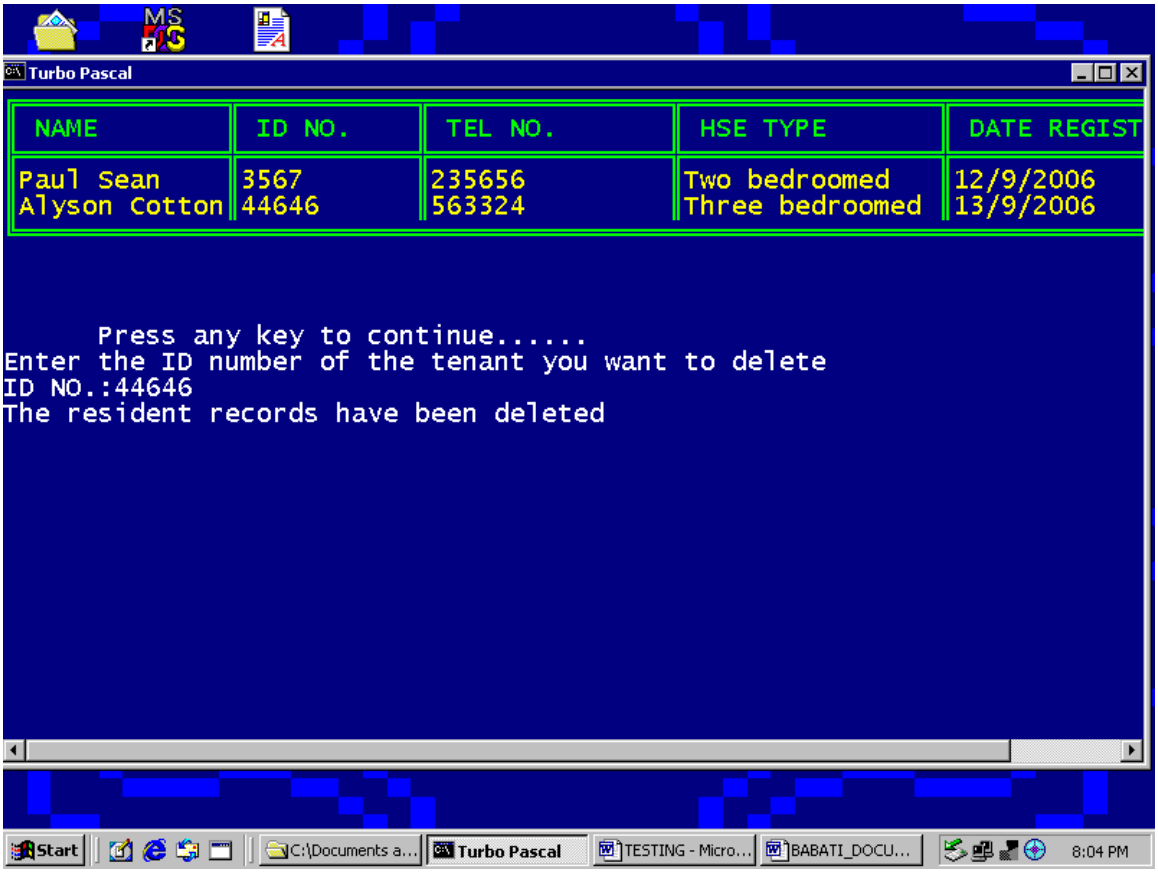
(d) Delete Testing

I) Entry of ID number that doesn't exist:



The program does not accept to delete since the ID number does not exist

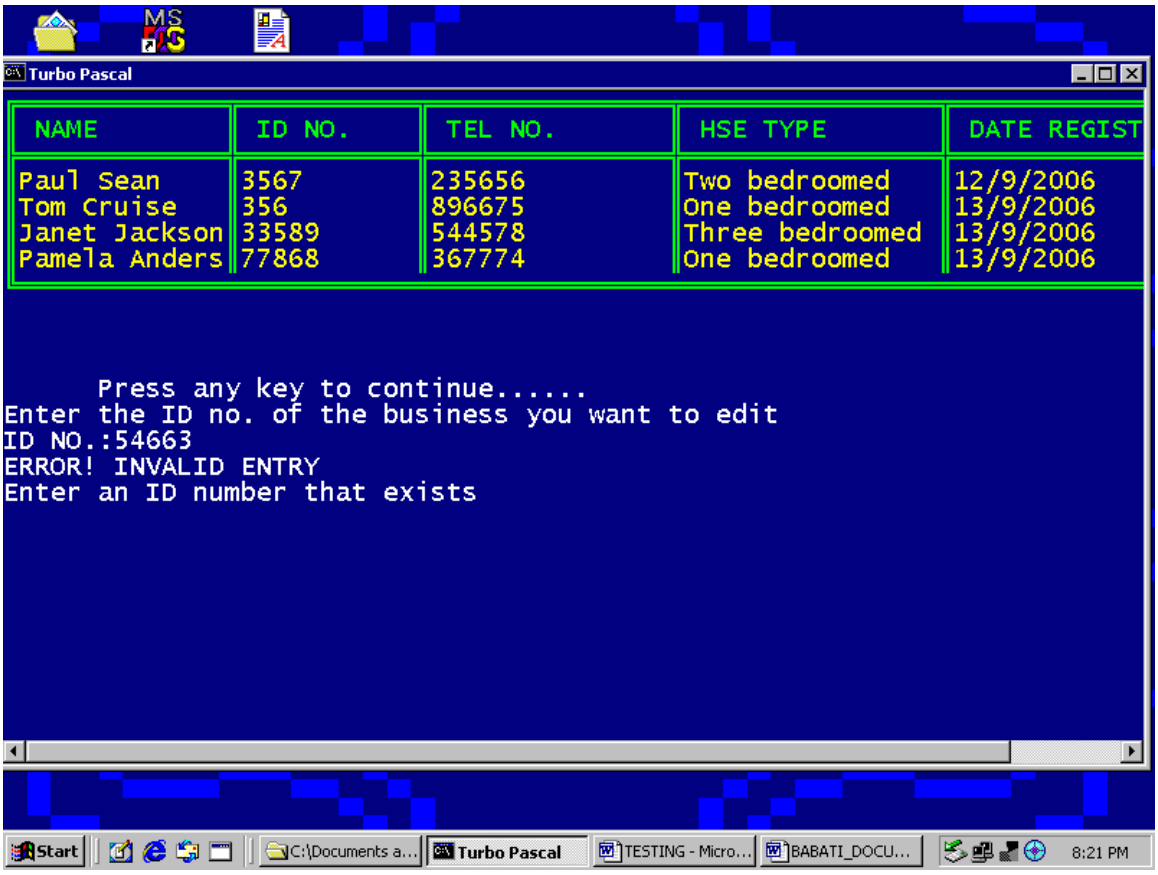
II) Entry of ID number that exists:



The program proceeds with deleting the records since the resident exists.

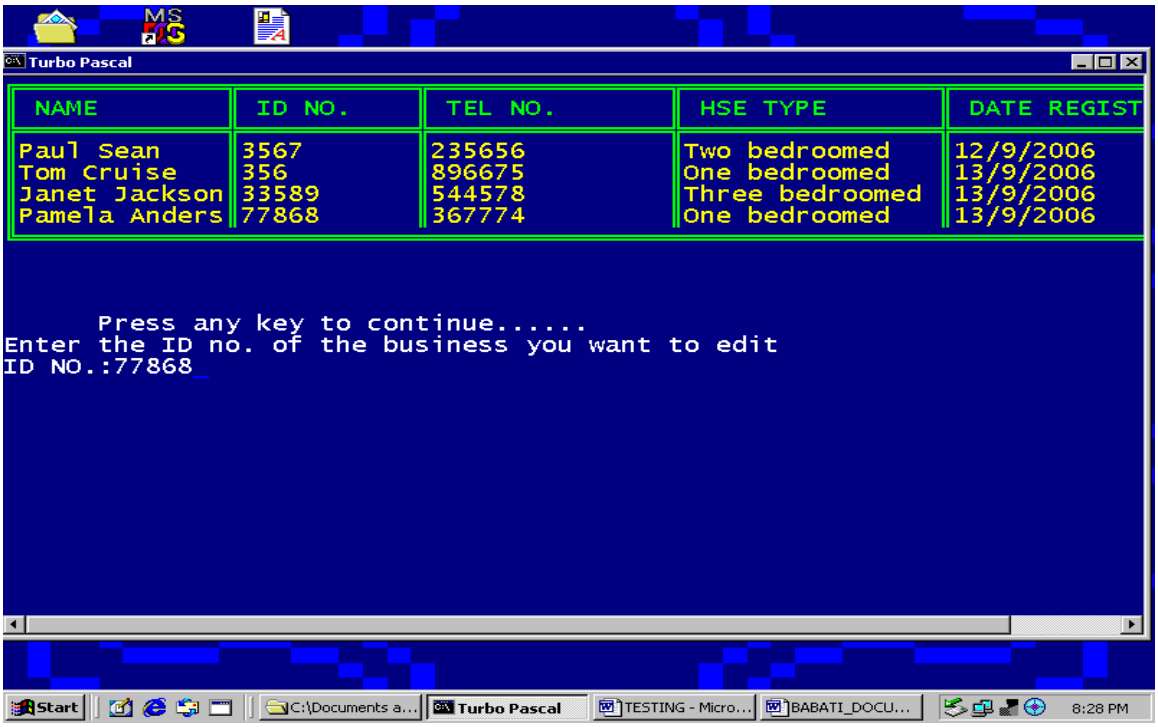
(e) Edit Testing

I) Entry of ID number that doesn't exist:

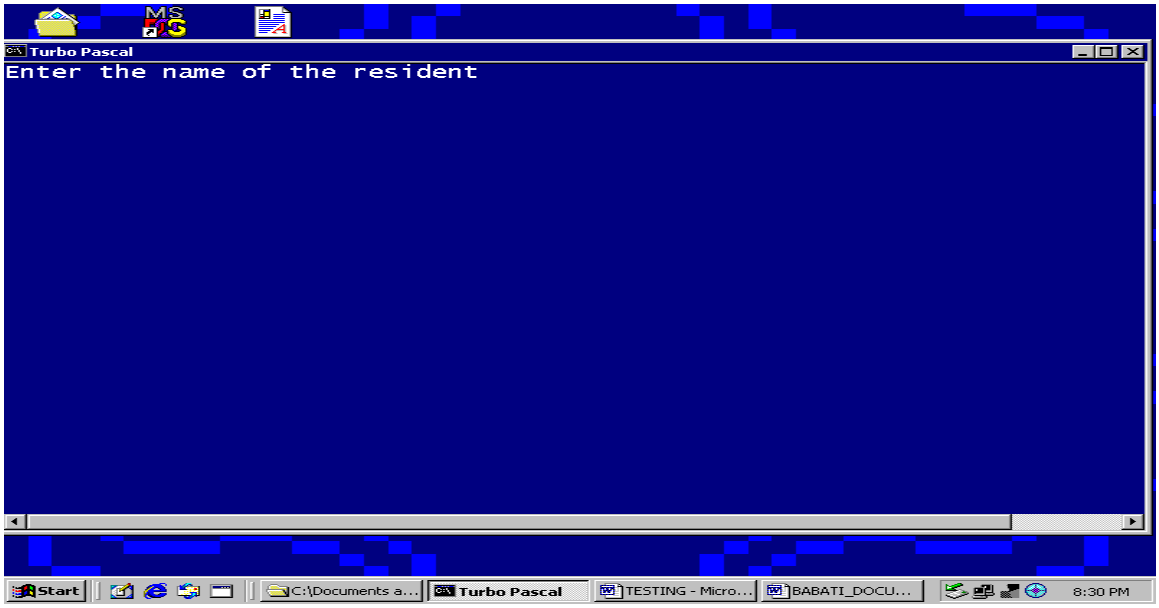


The program will not accept to edit the resident's details.

II) Entry of ID number that exists:



The program accepts the ID number and continues with the editing.



7. USER MANUAL

(a) Installation

The user must install the program in the computer from a floppy diskette. The following steps are followed during installation of the program:

- Insert the floppy disk in the computer.
- Create a folder in drive (C:) and name it **Babati Urban Council.**
- Click on 'My computer' and click on drive (A:).
- Select the program named Babati and copy it by clicking "copy" from the edit menu.
- Paste the program in drive (C:) in the folder named Babati Urban Council by selecting the folder and click on "paste" from the edit menu.

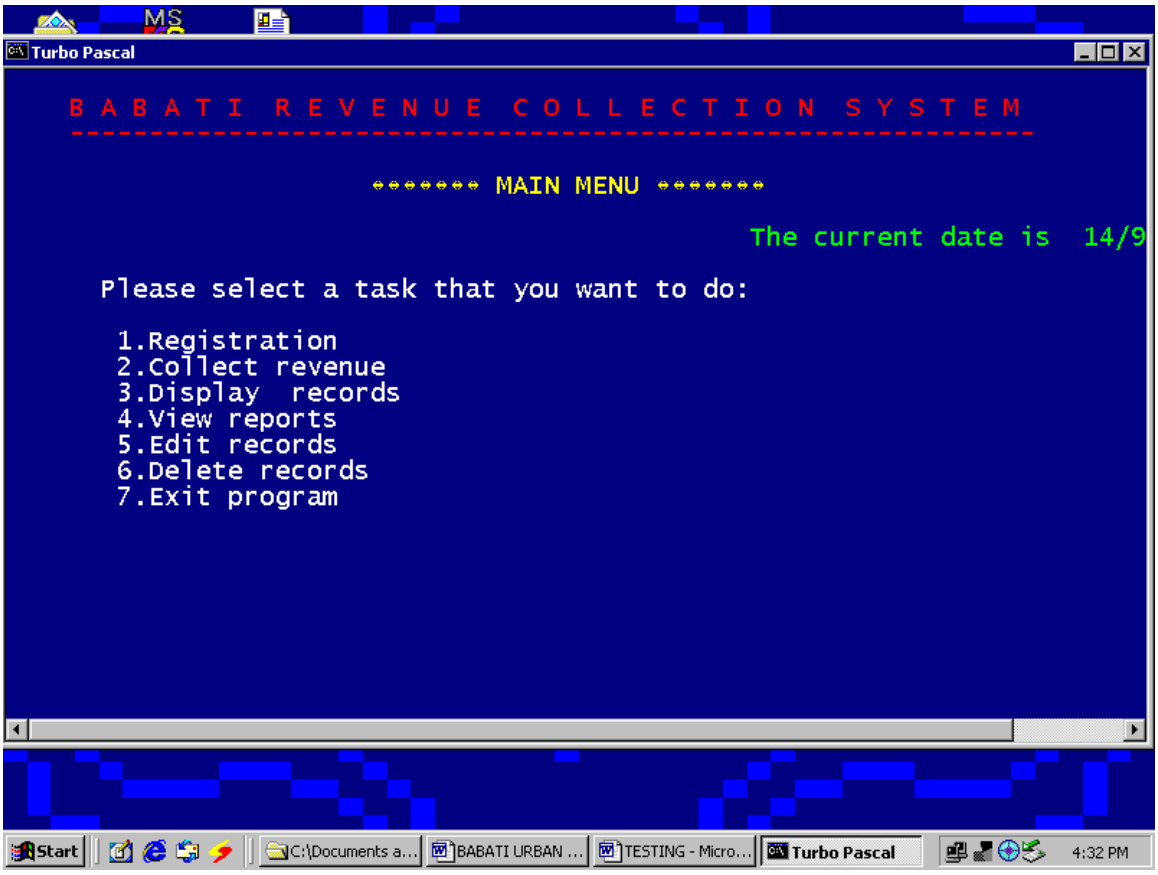
(b) Running the Program

To run the program, the user must do the following:
Start up the computer

- Open the folder containing the program by double clicking it's icon.
- Double click the program's icon.

(c) Operating the Program:

On running the program the main menu appears:



Registration

This is done by choosing the service that the user wants to register.
The services are:

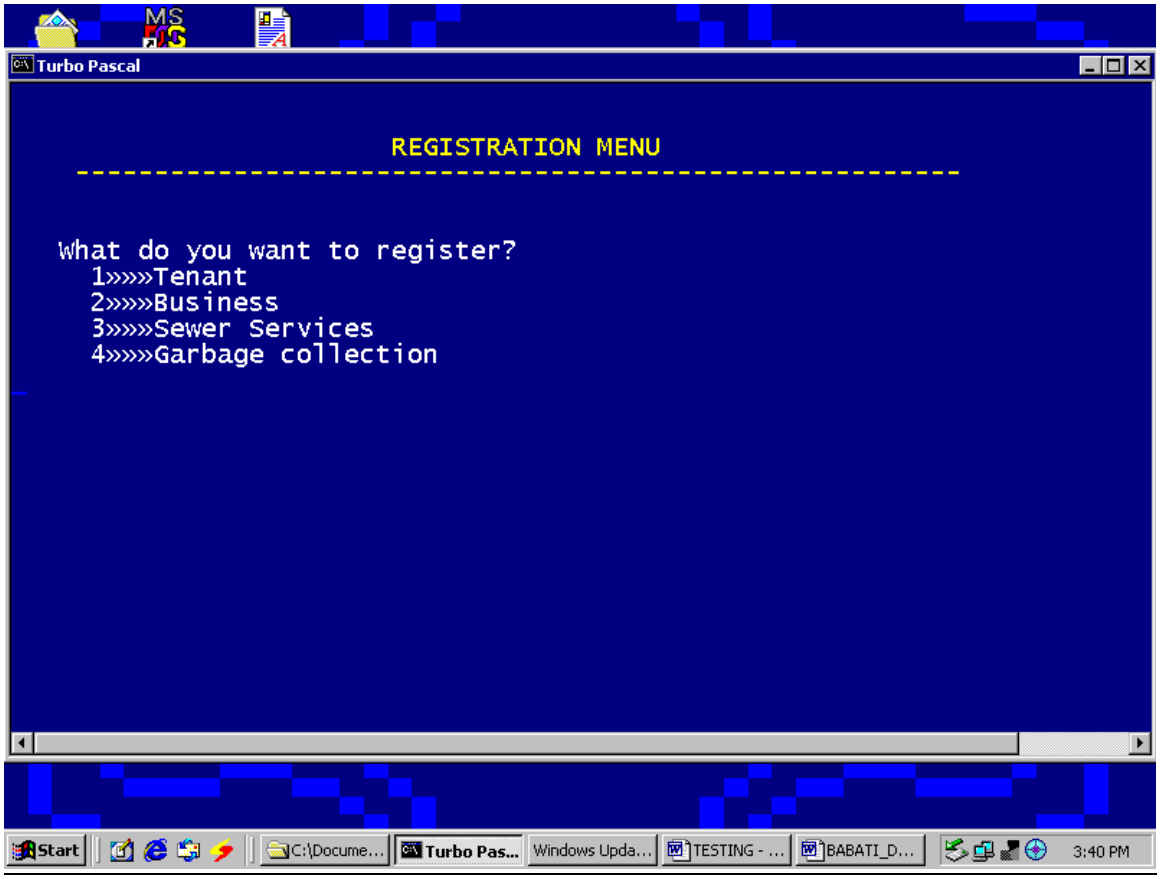
House rent: this deals with registering tenants who want to live under council houses. The residents can choose the type of houses they want which are: one bed roomed, two bed roomed, three bed roomed and four bed roomed.

Business licensing: this deals with registering residents who want to transact business within Babati Council. The types of businesses licensed are: general shop, hardware shop, private institution and miscellaneous business.

Sewer services: this deals with registering the tenants who want to be connected to the sewer line. There are two types of subscription, which are monthly and annual.

Garbage collection: this deals with registering all tenants to be catered for garbage collection.

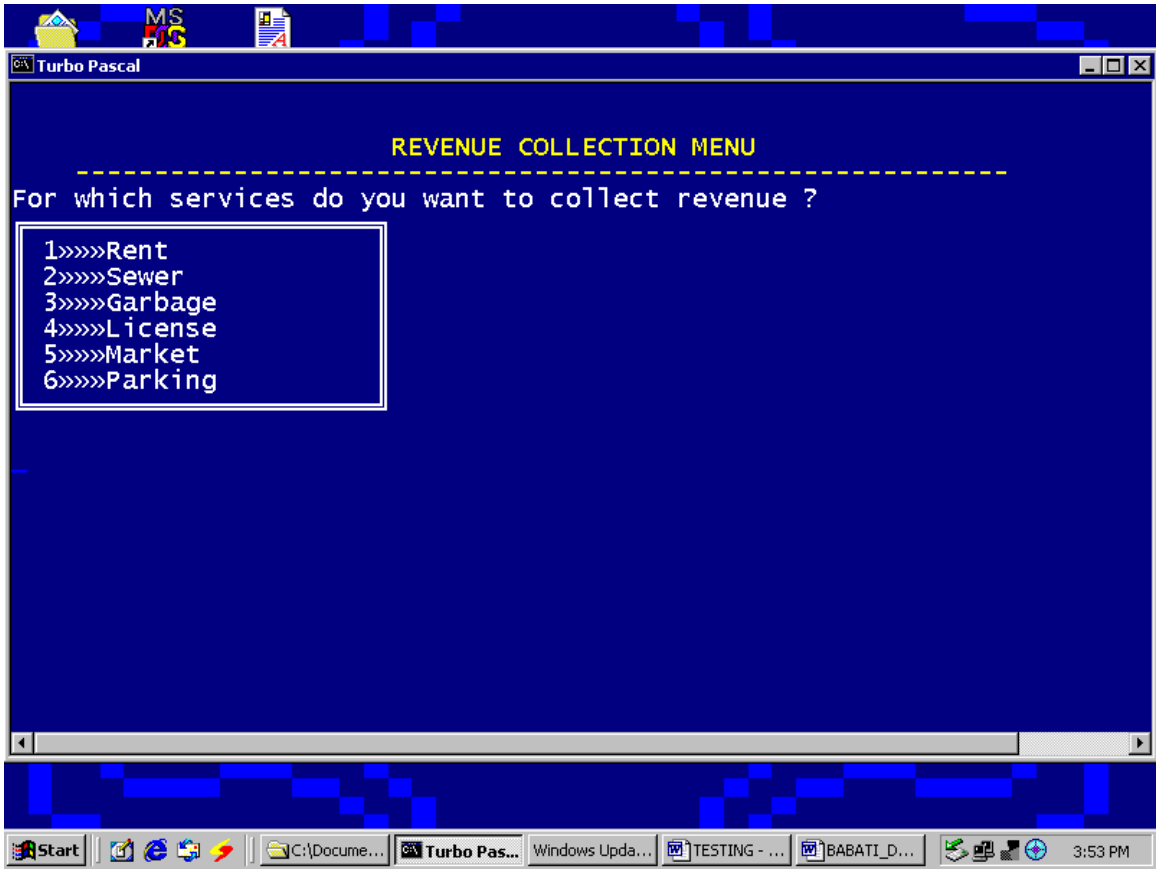
On selecting registration the following menu appears:



Collecting revenue

In collection of revenue the user collects payments made by residents. The user collects revenue for various services, which are housing, business license, sewer service and garbage collection.

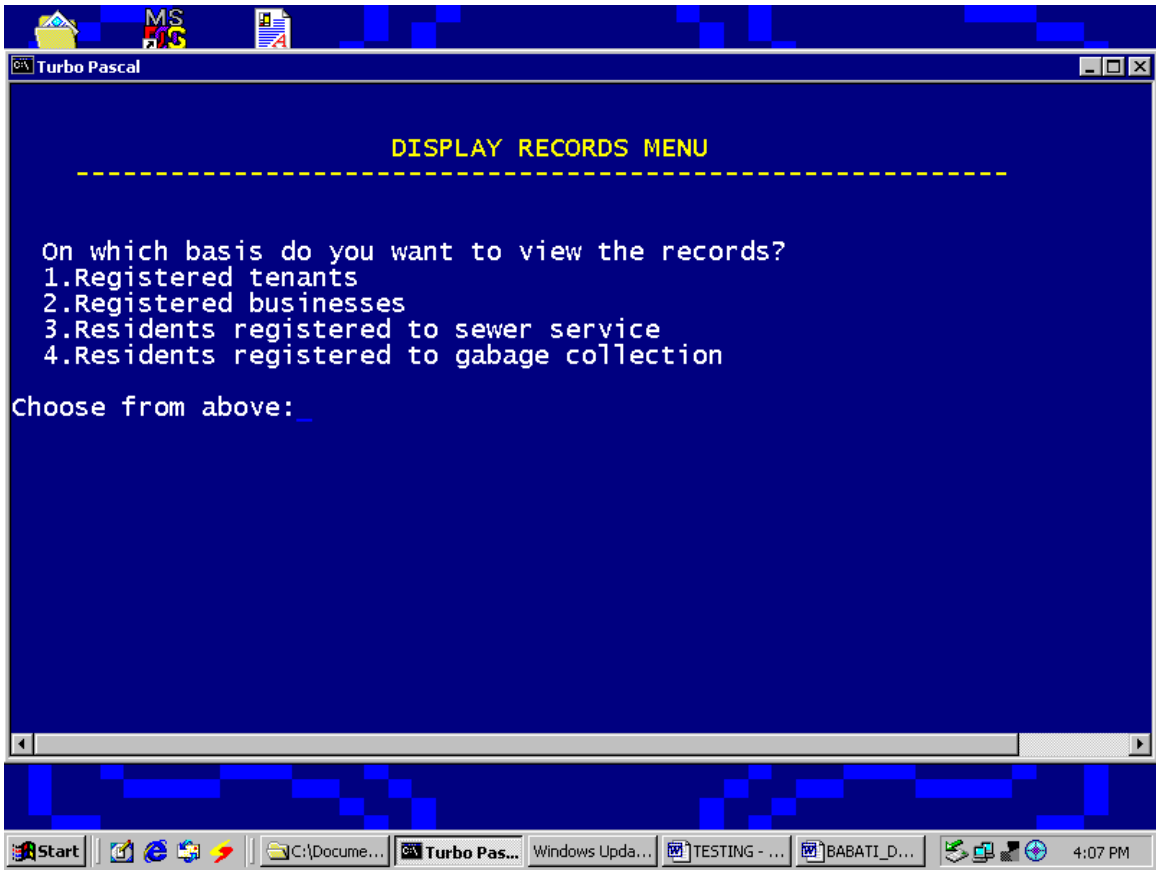
The following menu appears when collecting revenue is chosen:



Display records

In choosing the display records option the user can opt to view details of registered tenants, registered businesses, residents registered to the sewer services and garbage collection.

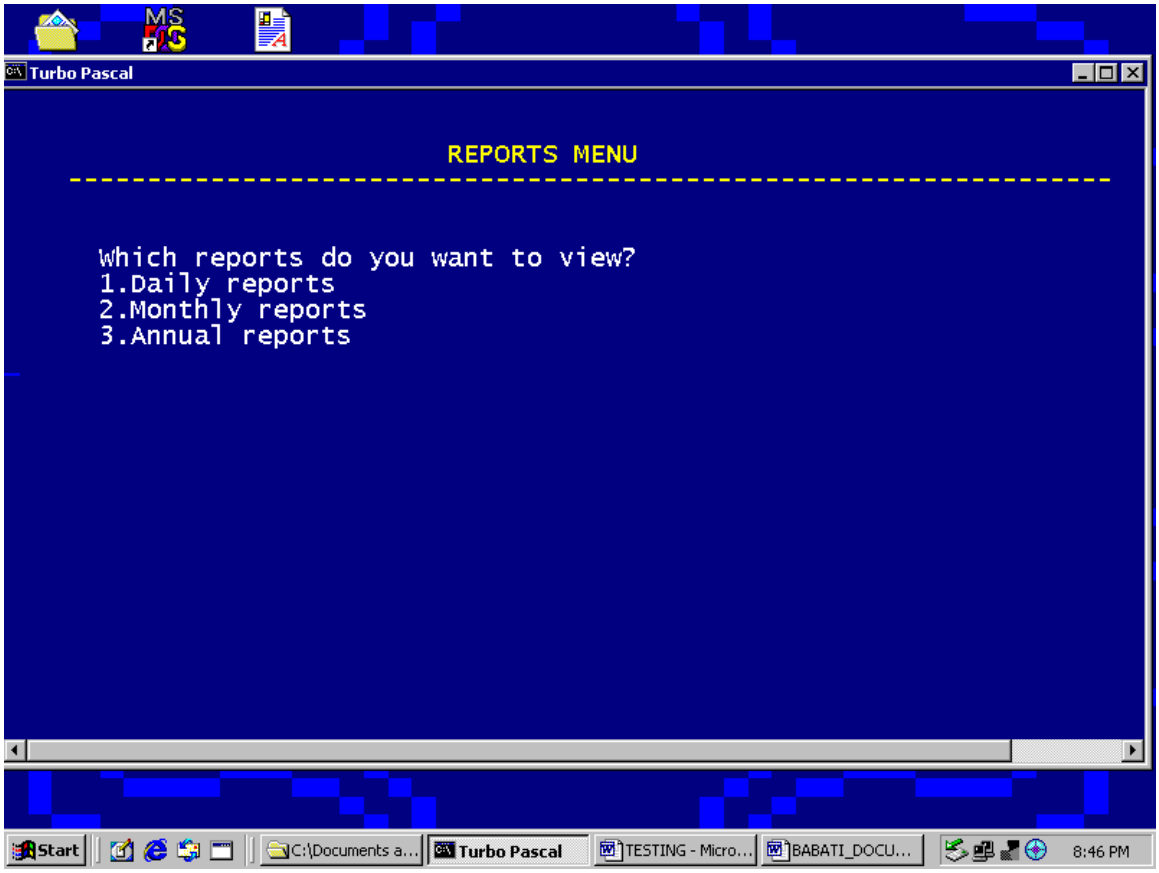
The following menu appears when display records is chosen:



View reports

On choosing this module, the user can view reports of revenue collected daily, monthly or annually from different services offered by the council. It also shows the total revenue collected.

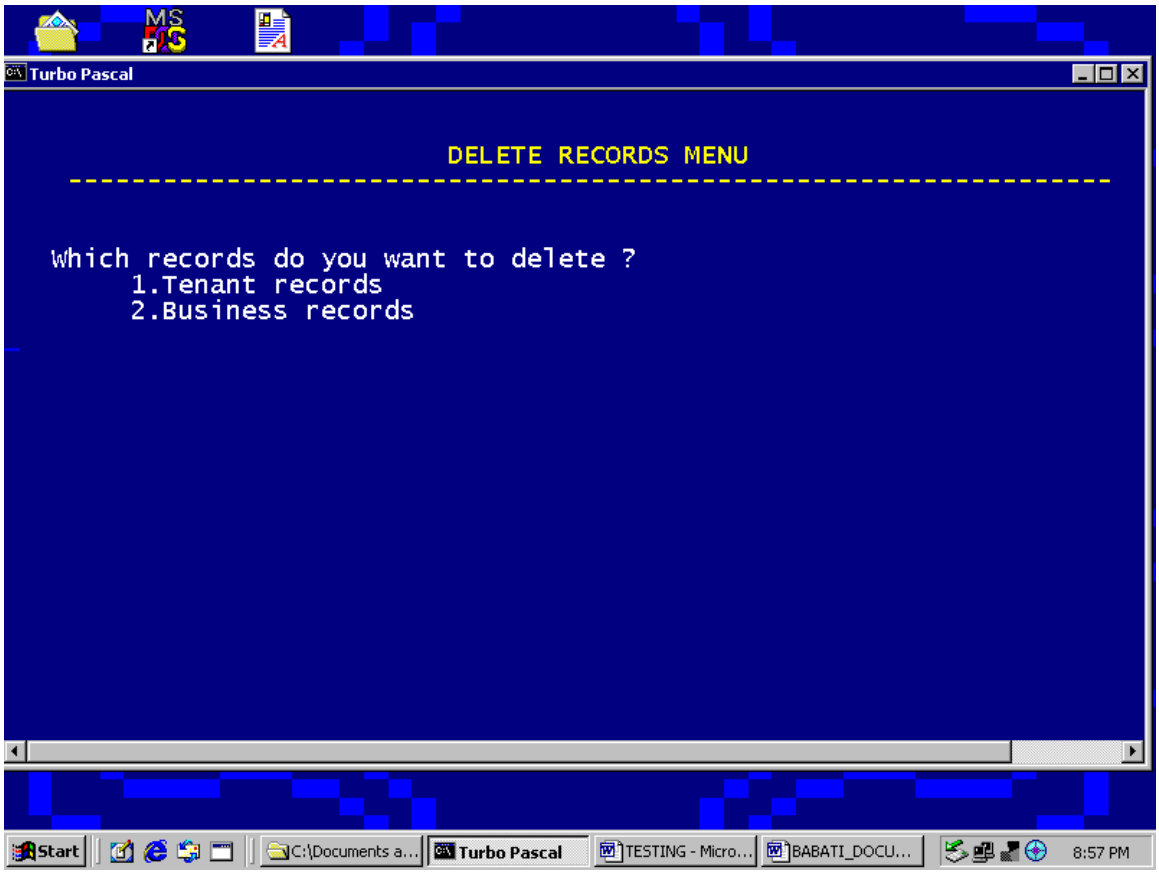
The following menu appears when choosing view reports:



Delete records

On choosing the delete records options, the user can delete tenant records or business records.

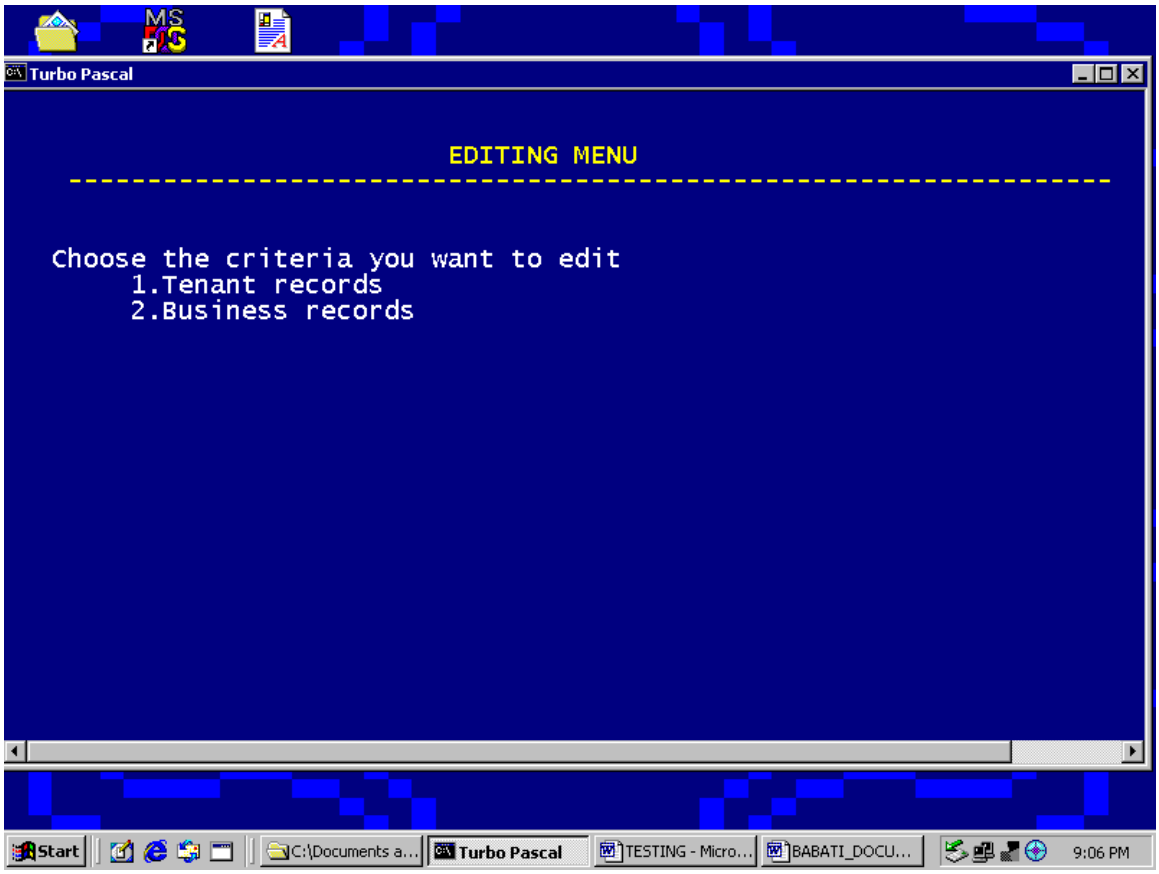
The following menu appears when choosing delete records:



Edit records

On choosing the edit records options the user can edit tenant and business records and input new details about the residents.

The following menu appears on choosing edit records:



8. CONCLUSION

In conclusion, it is hoped that the computerized revenue collection system will replace all manual revenue collection systems of Babati Urban Council that are carried out smoothly and increase speed and efficiency of processing data.

Developing of the program has been a good learning experience, which I have highly appreciated. Through this, I have learnt various programming concepts, which are useful in all spheres of programming.

9. RECOMMENDATION

Due to limited memory space on the floppy diskette, it is recommended that as soon as Babati Urban Council revenue collection system is procured, it should be installed.

For best results when using the revenue collection system, the program should be installed in a computer with the following specifications:

- At least 1 Gigabyte of hard-disk space.
- 3.5” floppy disk drive
- 1.2 Megahertz microprocessor
- 64 Megabytes of RAM (main memory)
- Operating System of Windows 95, 98, 2000, ME or XP.

It is also recommended that the revenue collection system can be used with other urban councils for revenue collection.

10. BIBLIOGRAPHY

During development of the program, reference was made to the following books:

- Holmes, B.J. *Pascal Programming*. The Guernsey Press Co Ltd., Great Britain.2000.
- Onunga John. *Introduction To Microcomputers And Programming.* Information System Academy Limited, Nairobi, Kenya. 1999.

11. APPENDICES

11.1.Pascal programming

Reserved words used

And	Links two arguments in logical comparison that must both be true to give a result
Begin	Marks the start of a block code.
Case	Marks the beginning of **case…do…** structure.
Do	Used in conjunction with **for** and **while**
Else	Optional extension to the **if** and **case** statements to handle those Sections not covered by positive selections.
End	Marks the end of a begin block, case or record structure.
File	Structured data type to define a variable that handles a link to a disk file.
For	Marks beginning of a **for…do…** structure.
Function	Used to introduce a function declaration.
If	Marks the beginning of the **if…then** or **if …then…**else control Structure.
Not	Operator that reverses the truth of a logical expression.
Of	Use in array, set and file type declarations, and in the case Statement.
Or	Links two arguments in a logical test. Either or both of the arguments are true.
Procedure	Marks the start of a procedure declaration.
Program	Marks the start of a program.
Record	Used in defining record structures in the type definition.
Repeat	Marks the start of a repeat until structure.
String	Declares a string variable. A string is an array of characters.
Then	shows statements to be performed if an if condition proves to be true.
To	Marks the upper limit of a for statement.

Type	Introduces user defined type definitions.
Until	The bottom line of a repeat loop, holding the exit test.
Uses	Declares the units that are going to be used in the program.
Var	Marks the beginning of variable declarations.
While	Marks the beginning of a while…do… control structure.

Standard procedures used

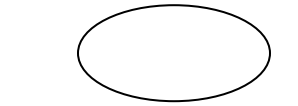
Assign	Assigns the name of an external file to a file variable.
Bar	Draws bar using current fillstyle and fill color.
Close	Closes an open file.
Close graph	Shuts down graphic system.
ClrScr	Clears the screen and returns cursor to upper left corner.
Delay	Delays a specific number of milliseconds.
Detect graph	Checks the hardware and determines which graphic driver and mode to use
Getdate	Returns current date set by the operating system.
Gotoxy	Moves cursor to given coordinates within virtual screen.
Initgraph	Initiates graphic system and puts the hardware into graphics mode.
Outextxy	Sends a string to the output device.
Read	reads a file component into a variable.
Readln	Execute the read procedure the skips to the next line of the file.
Rewrite	Creates and opens a new file.
Seek	Moves a current position of a file to a specified component.
SetBkColor	Sets the background color using the palette.
SetColor	Sets the current drawing color using the palette.
TextBackground	Selects the background color.
Textcolor	Selects the foreground character color.
Window	Defines a text window on a screen.
Write	Writes a variable into a file component or one or more variable into a file.
Writeln	Executes the write procedure, then outputs an end-of-line marker

the file.

Starndard functions used

Eof	Returns the end-of-file status.
Exit	Exits immediately from current block.
FilePos	Returns the current position of a file.
FilesSize	Returns the current size of a file.
IOResult	Returns the status of the last I/O operation performed.
Truncate	Truncate a file at the current file position.
Upcase	Converts a character into upper case.
WhereY	Returns the y-coordinate of the current cursor location.

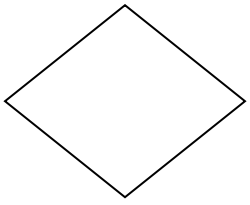
11.2.Flowchart symbols used



Start/end

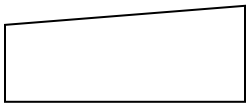
Input/output

Process

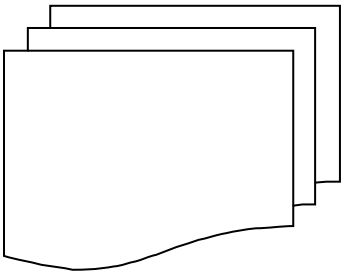


Decision

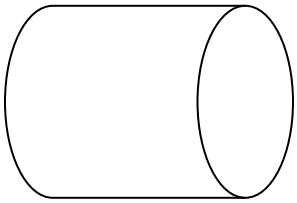
Manual Operation



Manual input



Multiple documents



Direct disk access

Display