ภาคผนวกที่ 2

เป็นตัวอย่างโปรแกรมเพื่อช่วยงานในร้านซุปเปอร์มาร์เก็ตทั่วไป สามารถแก้ปัญหาในการซื้อ และ ขายสินค้าของลูกค้า สามารถช่วยเจ้าของร้านเพื่อสั่งซื้อสินค้าที่ใกล้หมด อีกทั้งช่วยในการตรวจสอบยอดสินค้าคงเหลือรวมทั้งคิดกำไร ขาดทุน ได้อีกด้วย

```
        {       ////////ramkhamheang  university
                by  mr boonchok  jiamsawatphunt  385003002 ********
                22/1/40   }
program     project;
uses crt,dos,printer;
const
        taxid = '3-1011723-62';
        shopname = 'ram minimart';
        max_1 = 50;
        max_2 = 50;
type
       responsetype  = (noresponse,arrow,key,return);
       movement     = (none,left,right,up,down);
      menurec = record
             numchoices  :integer;
             menuwidth   :integer;
             choices:array[1..9] of char;
             descriptions:array[1..9] of string[80];
             title       : string;
             prompt   : string;
        end;
     recmaster = record
             id    : string[5];
             des   : string[10];
             onhand     : integer;
             cost_price  : real;
             sell_price   : real;
             day_import  : string[7];
         end;
      indexrec = record
         id                  : string[5];
        relative_address     : integer;
       end;
       summaryrec = record
        {  objective - calculate profit/perday }
               id            :string[5];
               name          :string[10];
               totalsell      :real;
               totalgoods      :integer;
               profitperonce :real;
               {  collect  order of good sell/perday   }
           end;
        filemaster = file of  recmaster;
        fileindex   =  file of  indexrec;
        filetran      =  file of  recmaster;
        filesumma =  file of  summaryrec;
        group_1 = array[1..max_1]  of recmaster;
        group_2 = array[1..max_1]  of indexrec;
        group_3 = array[1..max_2]  of summaryrec;
  const
         { create menu}
         mainmenu    :menurec = (
                    numchoices:9;
                    menuwidth:47;
                    choices:'12345678e';
          descriptions :('c reatefile',
                           'e nterdata',
                           'r eadallfile',
                           's earch',
                           's ellgood',
                           'u pdatedata',
                           's ummary',
                           'e dit data',
                           'e xit');
             title      :' supermarket  program ';
             prompt :'  enter choice  uses arrow key and press <enter>');
        no”              = #7;
        back_space       = #8;
        carriage_return  = #13;
        special          = #0;
        right_arrow      = #77;
        left_arrow       = #75;
        down_arrow       = #80;
        up_arrow         = #72;
        f1 = 'a:\master.dat';
        f2 = 'a:\transaction.dat';
        f3 = 'a:\index.dat';
        f4 = 'a:\summary.dat';
   var
     fp_m   :  filemaster;
     fp_i      : fileindex;
     fp_t      : filetran;
     fp_s      : filesumma;
     response  :char;
         {--- ------------ startsound ---------------------}
         procedure startsound;
         begin
           sound(1000);  delay(100);        sound(1300);  delay(100);
           sound(700);   delay(100);        sound(1500);  delay(100);
           sound(500);   delay(100);        sound(450); delay( 100);
           nosound;
        and;
         {............... ticsound______________________}.
         procedure ticsound;
         begin
           sound(4550); delay( 100);           sound(850);  delay(500);
           sound(1650); delay(100);            sound(1000); delay(100);
           nosound;
         end;
        {...................................................}
   procedure  setvideo(attribute:byte);
   var
       blinking,
            bold :byte;
   begin
      blinking:=(attribute and 4) * 4;
      if (attribute and 1) = 1 then
        begin
          bold:=(attribute and 2)*4;
          textcolor(blinking + bold);    textbackground(1);
        end
      else
        begin
          bold:=(attribute and 2) * 5 div 2;
          textcolor(5 + blinking + bold);     textbackground(0);
        end;
 end;
{-------------------------------------------------------}
procedure    dosdatetime;
var yr,mo, day, dayofwk    :word;
     hr,min, sec,hundredths :word;
     procedure     writeday(index:word);
      begin
       textbackground(6);       textcolor(10+128);
       case index of
       0:write('sunday :');
       1:write('monday :');
       2:write('tuesday :');
       3:write('wednesday :');
       4:write('thursday :');
       5:write('friday    :');
       6:write('saturday :');
       else
       write('day unknow ');
       end;
       end;
      begin
        getdate(yr,mo,day,dayofwk);          gettime(hr,min,sec,hundredths);
        highvideo;                           textcolor(4+128);
        setvideo(6);                         writeday(dayofwk);
        textcolor(15+128);                   writeln(mo,'-',day,'-',yr);
        textcolor(14+128);                   write('time     :');
        textcolor(10+128);                   write(hr,':',min,':',sec,':',hundredths);
        readln;textbackground(0);
      end;
       {-----------------------------------------}
       procedure  putstring(outstring:string;line,col,attrib:integer);
        begin
          {set video attribute and cursor location}
          setvideo(attrib);                gotoxy(col,line);
          highvideo;                       write(outstring);writeln;
          {reset nornal video}
       end;
    {--------------------------------------------}
     procedure   putprompt(outstring:string;line,col:integer);
      begin
        gotoxy(col,line);                 clreol;
        putstring(outstring,line,col,3);
      end;
    procedure  putcenterstring(outstring:string; line,attrib:integer);
       begin
         putstring(outstring,line,40 - length(outstring) div 2,attrib);
       end;
    {-----------------------------------------------------}
    procedure  getresponse(var response   :responsetype;  var direction  :movement;
                           var keyresponse:char);
       var
          inchar:char;
         begin
          response:=noresponse;   direction:=none;   keyresponse:=' ';
          repeat
            inchar:=readkey;
            if inchar = special then
     begin
      response:=arrow;
      {arrow are actually two character responses}
     inchar:=readkey;
      if inchar = left_arrow then                  direction:=left
           else
            if inchar = right_arrow then           direction:=right
           else
            if  inchar = down_arrow then           direction:=down
           else
            if inchar = up_arrow then              direction:=up
          else
            begin
             response:=noresponse;                 write(bell);
            end
        end
        {check for carriage returns}
        else
         if inchar = carriage_return then          response:=return
        {check for key entered,set to uppercase}
        else
        begin
         response:=key;                             keyresponse:=upcase(inchar);
        end;
        until response <> noresponse;
  end;
  {**************************************************}
procedure  displayframe(numlines,numcols:integer);
var
   leftcol,  topline,  col,line:integer;
 begin
  {limit the frame size to 20x73}
  if numlines > 20 then   numlines:=30;
  if numlines > 73 then    numcols:=78;
  highvideo;setvideo(6);  topline:=12 - numlines div 2 -3;
  if topline <1 then      topline:=1;
  leftcol:=40 - numcols  div 2 -3;
```

```
        if leftcol < 1 then     leftcol:=1;
        {display the top line}
        gotoxy(leftcol,topline);  write(' ',#218);
        for col:= 1 to numcols + 2 da write(#196);    write(#191,' ');
        (display the second line}
        gotoxy(leftcol,topline+1);  write(' ',#179,' ');
        for col:= 1 to numcols do  write(#220);  write(' ',#179,' ');
        for line:= topline + 2 to   topline + numlines + 2 do
         begin
            gotoxy(leftcol,line); write(' ',#179,' '); gotoxy(leftcol+numcols+3,line); writeln(' ',#179,' ');
         end;
         {display the second-last line}
         gotoxy(leftcol,topline+numlines+3);  write(' ',#179,' ');
         for col:= 1 to numcols do  write(#223); write(' ',#179,' ');
         {dislay the last line}
         gotoxy(leftcol,topline+numlines+4);  write(' ',#192);
         for col := 1 to numcols+2 do write(#196);    write(#127,' '); setvideo(6);
    end;
     {*******************************************************}
     procedure  displaymenu(menu:menurec);
     var
         leftcol,  topline,   spacing,i:integer;
      begin
         with menu do
           begin
             if numchoices > 7 then   spacing:= 1    else   spacing:=2;
              topline:=10 - (numchoices*spacing) div 2 - (2-spacing)* numchoices mod 2;
              leftcol:=41 - menuwidth div 2;
              {display the frame and title}
              displayframe(numchoices*spacing + 3,menuwidth); putcenterstring(title,topline+1,2);
              [display the choice and descriptions}
               for i:= 1 to numchoices do
                 begin
                   putstring(choices[i],topline  +i*spacing+2,  leftcol+2,2);
                   putstring('-'+ descriptions[i], topline + i*spacing +2,leftcol+4,0);
                 end;
                  {display the prompt}
```

```
                    putprompt(prompt,topline + numchoices*spacing + 7,  leftcol -2);
                   writeln;writeln;
          end;
   end;
    {*******-------------------------------------------********}
    procedure  getmenuresponse(menu:menurec;  var userchoice:char);
    var
       currentchoice,  spacing,  firstline,  desccol,  i   :integer;
       resp:responsetype;
       dir :movement;       {arrow key response}
       found:boolean;
   begin
     with menu do
     begin
     { determine the current(highlighted) choice}
     currentchoice:=numchoices;
     for i:= 1 to numchoices do
            if userchoice = choices[i] then         currentchoice:=i;
     if numchoices > 7 then   spacing:=1      else spacing:=2;
     firstline:=12 - (numchoices*spacing) div 2-   (2-spacing)*numchoices mod 2;
     desccol:=47 - menuwidth div 2;
     repeat
     {highligth current choice}
     putstring(descriptions[currentchoice], firstline + currentchoice*spacing,desccol,6);
     gotoxy(desccol - 8 + length(prompt),  firstline + numchoices*spacing + 5);
     getresponse(resp,dir,userchoice);           putstring(descriptions[currentchoice],
     firstline + currentchoice*spacing,desccol,0);
     case resp of
     arrow:if (dir = down) and   (currentchoice = numchoices)  then  currentchoice:=1
           else     if dir = down then    currentchoice:=currentchoice+1
           else     if (dir = up) and (currentchoice = 1)  then  currentchoice:=numchoices
           else     if dir = up then    currentchoice:=currentchoice -1;
     key: begin
```

```
                                              currentchoice:=i;
                                            end;
                                     if found then  resp:=return  else   write(bell);
                                     end;
{if  carriage return , use current choice}
return:userchoice:=choices[currentchoice];
end;
until resp = return;
{re -highlight chosen choice}
putstring(descriptions[currentchoice],firstline + currentchoice*spacing,desccol,6); writeln;
end;
end;
 {*************** ----------------------------.************}
  procedure createfile;
   var ch:char;
      i:byte;
              (* 1 ++++++++++++++++*)
              procedure createm;
               begin
                 assign(fp_m,f1);    rewrite(fp_m);  close(fp_m);
               end;
               (*2 +++++++++++++++++*)
              procedure  createtran;
                begin
                  assign(fp_t,f2);      rewrite(fp_t);    close(fp_t);
                end;
                (*3 +++++++++++++++++++ *)
                procedure  createin;
                  begin
                    assign(fp_i,f3);   rewrite(fp_i);     close(fp_i);
                  end;
                 (* 4 +++++++++++++++++++*)
                procedure  createsum;
                begin
                 assign(fp_s,f4);       rewrite(fp_s);    close(fp_s);
                end;
```

```
begin     clrscr;
 repeat
 textbackground(0);      clrscr;
 for  i := 1 to 7 do writeln;
 gotoxy(15,3);write(chr(201));
 for  i:= 1  to 40  do  write(chr(205));
 write(chr(187));gotoxy(15,4);              write(chr(186));textcolor(1);
 highvideo;write('enter choice ':25);        setvideo(6);gotoxy(56,4);
 writeln(chr(186));setvideo(6);              gotoxy(15,5);write(chr(186));
 gotoxy(56,5);write(chr(186));               setvideo(6);gotoxy(15,6);
 write(chr(186));textcolor(14);              highvideo;gotoxy(20,6);
 write('1','  create mester file ');          setvideo(6);gotoxy(56,6);writeln(chr(186));
setvideo(6);gotoxy(15,7);write(chr(186));  textcolor(14);highvideo;gotoxy(20,7);
 write('2','       transaction file');          setvideo(6);gotoxy(56,7);
 writeln(chr(186));setvideo(6);              gotoxy(15,8);write(chr(186));
 textcolor(14);highvideo;                    gotoxy(20,8);write('3','      index file');
 setvideo(6);gotoxy(56,8);writeln(chr(186)); setvideo(6);gotoxy(15,9);
 write(chr(186));textcolor(14);              highvideo;gotoxy(20,9);
 write('4','       summary  fib');              setvideo(6);gotoxy(56,9);writeln(chr(186));
 setvideo(6);gotoxy(15,10);write(chr(186));  textcolor(14);highvideo;gotoxy(20,10);
 write('5',      all   ');  setvideo(6);gotoxy(56,10);  write(chr(186));setvideo(6);
 gotoxy(15,10);write(chr(186));  setvideo(6);gotoxy(15,11);  write(chr(186));textcolor(14);
 highvideo;gotoxy(20,11);  write('6','  exit');  setvideo(6);gotoxy(56,11);writeln(chr(186));
 setvideo(6);gotoxy(15,12); write(chr(186));textcolor(red);highvideo;
 gotoxy(25,12);write('choose :-> ');    setvideo(6);gotoxy(56,12);writeln(chr(186));
 setvideo(6);gotoxy(15,12);write(chr(200));
 for  i:= 1  to 40  do  write(chr(205));
 setvideo(6);writeln(chr(188));highvideo;   ch:=readkey;
   case   ch   of
    '1':createm;                  '2':createtran;
    '3':createin;                 '4':createsum;
    '5':begin
          createm;                 createtran;
          createin;                createsum;
       end;
     else
       begin
```

```
             if (ch <> '1') and  (ch <> '2') and  (ch <> '3') and  (ch <> '4') and
                (ch <> '5') and  (ch <> '6')
           then  begin
                   gotoxy(15,7);    writeln('invalid please try again ');
                 end;
          end;
        end;
    until (ch = '6') or( ch = '5');
    textbackground(0);clrscr;
  end;
   {--------------------------------------------}
   procedure     enterdata;
   const
         max = 50;
   type
        st = string[5];    grouparr = array[1..max] of st;
   var   a:grouparr;                   input:recmaster;
        keyinput:indexrec;              temp:st;
        test:boolean;                   num,i,j:integer;
    begin
      for num:= 1  to max do  a[num]:='0';
      textcolor(lightcyan);highvideo;   clrscr;num:=0;
      assign(fp_m,f1);reset(fp_m);  write('enter  id [end = 0]:');      readln(input.id);
        while   input.id  <> '0' do
            begin
              inc(num);      a[num]:=input.id;
             with input  do
              begin
                write('enter   description  :');     readln(des);
                write('        onhand :');            readln(onhand);
                write('        cost price :');        readln(cost_price);
                write('        sell  price :');       readln(sell_price);
                write('        day import  :');       readln(day_import);  write(fp_m,input);
               end;
                 write('enter   id [end = 0]:');      readln(input.id);
          end;
        close(fp_m);
```

```
         { sort  index  }
         i:=1;              test:=true;
           while (i < num) and test do
            begin
              test:=false;
          for j:= 1 to num - i do
                  begin
                    if (a[j] > a[j+1]) then
                      begin
                        temp:=a[j];  a[j]:=a[j+1];  a[j+1]:=temp;   test:=true;
                     end;
                end;    inc(i);
          end;
           { write data key id into index file }
        assign(fp_i,f3);   reset(fp_i);  j:=0;
           for  i:= 1 to num do
             begin
               keyinput.id:=a[i];            keyinput.relative_address:=j;
               write(fp_i,keyinput);        j:=j+1;
               end;     close(fp_i);    textcolor(white);
 end;
procedure     readallfile;
  var    i:byte;                         indexdata:indexrec;
    begin
      clrscr; gotoxy(32,3); textcolor(lightgreen+128);
      highvideo;writeln(shopname); gotoxy(29,4);
      textcolor(yellow+128);writeln(taxid);gotoxy(10,6);
     for i:= 1 to  20 do write('+*+');        writeln;gotoxy(25,8);
     textcolor(red+128);write(chr(1));textcolor(magenta);
     highvideo;write(' read all   index  file ');  textcolor(red+128);writeln(chr(2));
     assign(fp_i,f3);reset(fp_i);
     while not eof(fp_i) do
       begin
         read(fp_i,indexdata);
         with indexdata do
           begin
             textcolor(white);highvideo;   writeln(' id = ':22,id);
```

```
                    writeln(' index = :':25,relative_address);      readln;
                end; {end with }
             end;   {end while }
             close(fp_i);
         end;
        {+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
       procedure  search;
       var choice:byte;
       procedure   searchwithid;
       var
             code:str                       temp:recmaster;
             found:boolean;                 l,h,i,md,num:integer;              arr:group_1;
         begin
           num:=0;            assign(fp_m,f1);            reset(fp_m);
            while not eof(fp_m) do
             begin
               num:=num+1;            read(fp_m,arr[num]);
            end;            close(fp_m);
        {sort  array of  master file  for search }
        { use bubble sort }
             l:=1;         found:=true;
                while (l <num) and found do
                 begin       found:=false;
                   for h:=1 to num - 1 do
                     begin
                      if arr[h].id >  arr[h+1].id then
                        begin
                          temp:=arr[h];  arr[h]:=arr[h+1];  arr[h+1]:=temp;  found:=true;
                        end;
                     end; {end for h}     l:=l+1;
                  end;
  { binary search }
clrscr;textcolor(lightcyan);highvideo;  write('enter id that  search [0 =end] :'); readln(code);
while  code <> '0' do
 begin
  l:=1;h:=num+1;  i:=1;found:=true;
  while found and (i<=num) do
```

```
          begin       md:=(l+h) div 2;
           if code = arr[md].id then  found:=false
              else
                begin  if code < arr[md].id then h:=md   else l:=md;  end; i:=i+1;
          end;
       if not found then
        begin
          with  arr[md] do
           begin           writeln(' id : ',id);    writeln(' description :',des);
                           writeln(' onhand   :',onhand);writeln(' cost price :',cost_price:6:2);
                            writeln(' sell price :',sell_price:6:2);  writeln(' day import :',day_import);
            end;
       end
      else
        writeln(' sorry I can"t fide');  write('enter id that  search [0 = end] :');  readln(code);
  end;  {end while}
end;
    {*******************************************************}
    procedure  searchwithrelativeaddress;
     var   data:recmaster;          n:integer;
      begin
        clrscr;  assign(fp_m,f1); reset(fp_m);  textcolor(lighteyan); highvideo;
        write('enter  number of  address [end = 0]:');   readln(n);
        while n <> 0 do
         begin   seek(fp_m,n-1);    read(fp_m,data);
          with  data do
           begin          writeln(' id : ',id);      writeln(' description :',des);
                          writeln(' onhand   :',onhand); writeln(' cost price :',cost_price:6:2);
                          writeln(' sell price :',sell_price:6:2);    writeln(' day import :',day_import);
          end;
      write('enter  number of  address [end = 0]:');  readln(n);
     end;     close(fp_m);
  end;
   begin   repeat
               clrscr;       textcolor(lightcyan);highvideo; gotoxy(25,6);
               writeln('1 search file   with   id key');
               gotoxy(25,7); writeln('2                        relative address ');
```

```
             gotoxy(25,8); writeln('3 exit'); gotoxy(28,9); write('choice = '); readln(choice);
             case choice of
                     1:searchwithid;          2:searchwithrelativeaddress;
                     else
                     if (choice  <> 1) and  (choice  <> 2) and (choice <> 3)  then
                             gotoxy(25,10); writeln('invalid please try again!');
     end;
     until choice = 3;
end;
  {**********************************************************}
   procedure sellgood;
   var
     choice:byte;
      {  read data form master file to transaction file  }
      {  step 1 read data to array
         "--" 2 sort array
         "--" 3 write data to transaction file}
     procedure  sell1;
      var      tranarr:group_1;      i,n:integer;          temp:recmaster;
       begin
       n:=0;     assign(fp_m,f1);      reset(fp_m);
       while not eof(fp_m) do
          begin       n:=n+1;        read(fp_m,tranarr[n]);        end;
       close(fp_m);
       { write data to transaction file }
       assign(fp_t,f2);       reset(fp_t);
       for i:= 1 to n do
       begin     temp:=tranarr[i];    write(fp_t,temp);        end;
       close(fp_t);
    end; {end procedure sell1}
     {++++++++++++++++++++++++++++++++++++++++++++++++++++++}
   { begin sell good  }
    {  step 1
    * read data form transaction file to array
      for use to binary search
     step 2  sell good      }
     procedure  sell2;
```

```
const list = 20;
type
  recordbill = record
          descrip :string[10];
           price1  :real;
          volumn   :integer;
          end;
  billarray = array[1..list] of recordbill;
var
 trana:group_1;  i,j,vol,l,h,md,k,max1:integer;  found,have:boolean;  code:string[5];
 select:byte;     change,tax,recieve,tot, pay,profits:real;   presentbill:billarray;
 listnumber,numgood :integer;  temp:recmaster;  summ1:group_3;  totl :real;
begin
  tot:=0.0;   totl :=0.0;
  { begin sellgood }
  max1:=0;    assign(fp_t,f2);     reset(fp_t);
     while not eof(fp_t) do
     begin    max1:=max1+1;   read(fp_t,trana[max1]);     end;
     close(fp_t);
     {  sort array of transaction file}
     { use bubble sort }
     i:=1;       found:=true;
     while (i<max1) and found do
     begin        found:=false;
      for j:=1 to max1-i do
       begin
         if trana[j].id >  trana[j+1].id then
             begin
              temp:=trana[j];  trana[j]:=trana[j+1]; trana[j+1]:=temp;  found:=true;
             end;
       end; {end for j}       inc(i);
     end;  {end while loop}
     clrscr;  textcolor(lightcyan);highvideo; listnumber:=0;  j:=0;  k:=0;
     writeln;writeln;   write('enter id of good [ 0 = end] : ');  readln(code);
     while code <> '0'   do
     begin     have:=true;
       while   have do
```

```
   begin
    for i:=1 to max1 do
     if code =   trana[i].id  then    have:=false;
      if  have then
       begin  write(' no ! have id please  try again  : ');    readln(code);  end;
   end;
   listnumber:=listnumber+1;  l:=1; h:=max1+1;  i:=1;  found:=true;
    while found and (i<=max1) do
     begin      md:=(l+h) div 2;
      if code = trana[md].id then      found:=false
           else
             !
             begin  if code < trana[md].id  then  h:=md   else    l:=md;    end;
       i:=i+1;
   end; { end while loop  found and i<=n }
if not found then
  begin
    write('enter  volumn   :');
    { number of good that want buy}
      readln(vol);
      {check inventary good}
      {check good if less than 30 then make buy add  good}
      if  trana[md].onhand <= 30  then
       begin
        write('inventory check  !!!!!! good not  ');writeln('  insufficient  you buy  add good  ');
        writeln(' buy  good  add  emergency ');
       end;
  if vol > trana[md].onhand then
      . begin
        writeln(' good don"t enough for sell ');
        writeln('volumn of good  inventory check have   =',trana[md].onhand);
        write('please enter insist again vol  :'); readln(numgood);vol:=numgood;
       end;
   trana[md].onhand:=trana[md].onhand - vol;   assign(fp_t,f2); rewrite(fp_t);
       for i:= 1 to max1 do
        begin   write(fp_t,trana[i]);     end;
        close(fp_t);       j:=j+1;
        with  presentbill[j]  do
```

```
                 { use for bill to customer when id = 0}
                  begin
                   descrip:=trana[md].des;  price1:=trana[md].sell_price;   volumn:=vol;
                  end; { end  with}
            pay:=vol*trana[md].sell_price; profits:=(trana[md].sell_price - trana[md].cost_price)*vol;
            tot:=tot + pay;  writeln('profits = ',profits:2:2);     k:=k+1;
            with summ1[k]  do
             begin
              id:=trana[md].id;name:=trana[md].des;totalsell:=tot;totalgoods:=vol;profitperonce:=profits;
             end;
          if j = 1  then
          begin
           textcolor(yellow+128);highvideo;  for i:=1 to 80 do   write('*');     writeln;
           writeln(shopname:38);    writeln('tax id ':25,taxid);    writeln;
           writeln(' no':2,'des':10,'price':15,'vol':20);
           for i:=1 to 80 do     write('-');writeln;       textcolor(lightcyan);
       end;      write(j:2);
      with  trana[md] do
      begin    writeln(des:10,sell_price:15:2,vol:20);     writeln;    end;
    end;{ end if not found}
    write('enter id of good [ 0 = end} : ');   readln(code);
   if code = '0' then
    begin
    assign(fp_s,f4);   reset(fp_s);     seek(fp_s,filesize(fp_s));
    for  i:= 1 to k do   write(fp_s,summ1[i]);
     close(fp_s);  textcolor(red);highvideo;  writeln('total ',tot:2:2,' bath');
tax:=tot * 0.07;  { tot = tax + total}   tot1:=tot1 + tot + tax;
writeln('total + vat 7 % **** ', tot1:2:2,' bath');  write(' recieve  :'); readln(recieve);
 writeln;  change:=recieve - tot1;  writeln(' change ****',change:5:2,' bath');
writeln; readln;  textcolor(lightcyan);highvideo;  writeln(' 1  want bill for customer ');
writeln(' 2  don"t want bill');  write('enter choice :');  readln(select);
if select = 1 then
 begin
  for i:= 1 to 77 do  write(lst,'*');  writeln(lst);   writeln(lst,shopname:38);
 writeln(lst,' tax id  ':25,taxid); writeln(lst);writeln(lst,' no':2,'des':10,'price':15,'vol':20);
 for i:=1 to 77  do   write(lst,'-');writeln(lst);     j:=0;
  for i := 1 to  listnumber do
```

```
        begin
         j:=j+1;     write(lst,j:2);
         with presentbill[i]  do        writeln(lst,descrip:10,price1:15:2,volumn:20);
         end;
       writeln(lst);writeln('total  ******',tot:2:2,' bath');writeln(lst,'  total + vat 7 %  *****',tot1:2:2,' bath');
       write(lst,'  recieve '); write(lst,recieve:2:2);writeln(lst);
       writeln(lst,'  change ******',change:5:2,'  bath');  writeln(lst);
       writeln(lst,'  ++++++  thank you  +++++++ ':25);
   end{ end if  select = 1}
    else
      begin
       textcolor(yellow+128);highvideo;
       for i:= 1 to 80 do  write('*');  writeln;  writeln(shopname:38);
      writeln(' tax id  ':25,taxid); writeln;  writeln(' no':2,'des':10,'price':15,'vol':20);
     for i:=1 to 80 do  write('-');writeln;  j:=0;   textcolor(lightcyan);highvideo;
     for i := 1 to  listnumber do
     begin
       j:=j+1;  write(j:2);
       with presentbill[i]  do     writeln(descrip:10,price1:15:2,volumn:20);
      end;
    writeln;  writeln('   total *****',tot:2:2,' bath '); writeln('   total + vat 7 % *****',tot1:2:2,' bath');
    write('  recieve '); write(recieve:2:2,' bath');writeln;writeln('   change ******',change:5:2,'  bath');
    writeln;       writeln('  +++++ thank you ++++  ':25);
  end;
   textcolor(white);highvideo;  readln;
 end;{end if code = '0'}
end;{end while loop}
end; { end procedure sell2}
   begin
     repeat      clrscr;
       textcolor(lightcyan); highvideo;gotoxy(20,3);  writeln('1  write  data  to  transaction file ');
       gotoxy(20,4);writeln('2  sell good ');gotoxy(20,5); writeln('3  exit ');gotoxy(24,6); write('enter :');
       readln(choice);
       case choice  of
        1:sell1;           2:sell2;
         else
          begin
        if (choice  <> 1) and (choice  <> 2) and (choice  <> 3)
         then  writeln('invalid  please try again ');
        end;
    end;
     until choice = 3;
 end;
   {****************************************************}
   procedure  updatedata;
    var
      tran:group_1;        num,i:integer;
    begin
     clrscr;
     gotoxy(35,8);textcolor(red);      highvideo;write('u');textcolor(green+128);
     highvideo;writeln('pdate data');  gotoxy(25,10);textcolor(blue);
     highvideo;write('u');textcolor(lightcyan+128); writeln('pdate  master file with transaction  file ');
     gotoxy(25,11);textcolor(yellow);       write('u');textcolor(lightred+128);
     writeln('pdate  master file per day  ');  gotoxy(35,13);textcolor(white+128);
     writeln(' please  wait        ');       gotoxy(25,15);textcolor(lightmagenta+128);
     for i:= 1 to 32 do write(chr(14));
     {open transaction file
      input data to  array tempory }
      num:=0;    assign(fp_t,f2);    reset(fp_t);
      while not eof(fp_t) do
       begin     num:=num+1;    read(fp_t,tran[num]);        end;
     close(fp_t);
    {write data  form transaction file   to master file  }
    assign(fp_m,f1);    rewrite(fp_m);
    for i:=1 to num do
    begin   write(fp_m,tran[i]);    end;
  close(fp_m);
end;
 {**************************************************}
  procedure summary;
    var
       choice:byte;            i:byte;
     {9*******************************9}
       procedure   profit1;
```

```
var
   a:group_3;       num,i:integer;    sum:real;   totsell1:real;     totvol:integer;
begin
  assign(fp_s,f4);  reset(fp_s);    num:=0;
  while not eof(fp_s) do
    begin  num:=num+1;       read(fp_s,a[num]);   end;
   close(fp_s);  sum:=0.0; totsell1:=0.0;   totvol:=0;
   for i:= 1 to num do
      begin      totsell1:=totsell1 + a[i].totalsell;
       totvol:=totvol + a[i].totalgoods;   sum:=sum + a[i].profitperonce;
      end;
  clrscr;
  textcolor(blue+128);highvideo;                  gotoxy(32,4);writeln(shopname);
  textcolor(yellow +128);highvideo;                gotoxy(29,5);writeln(taxid);
  textcolor(lightcyan);highvideo;                 gotoxy(13,7);for i:= 1 to  5 do
   begin    write(chr(3));                   write(chr(12));            end;
   textcolor(lightgreen);highvideo;
   for i:= 1 to 5 do
   begin    write(chr(14));     write(chr(2));    end;t extcolor(lightmagenta);highvideo;
   for i:= 1 to  5 do
   begin    write(chr(3));  write(chr(12));    end;    textcolor(lightgray);highvideo;
   for i:= 1 to 5 do
   begin   write(chr(14));  write(chr(2));   end;   textcolor(lightmagenta);highvideo;
   for i:= 1 to  5 do
   begin  write(chr(3));  write(chr(11));   end;  textcolor(lightred);highvideo;
 gotoxy(25,10);writeln('total  goods  is that   sell/day = ',totsell1:2:2,' bath');
 textcolor(lightgreen);highvideo;
 gotoxy(25,12);writeln('total  goods volumn is that  sell//day = ',totvol);
 textcolor(lightcyan);highvideo; gotoxy(25,14);
 writeln('profit per day = ',sum:2:2,' bath');   textcolor(lightcyan);highvideo;
 gotoxy(13,18);for i:= 1 to  5 do
  begin  write(chr(3));   write(chr(12)); end;textcolor(lightgreen);highvideo;
  for i:= 1 to 5 do
  begin  write(chr(14)); write(chr(2)); end; textcolor(lightmagenta);highvideo;
  for i:= 1 to  5 do
  begin write(chr(3));  write(chr(12));  end;     textcolor(lightgray);highvideo;
  for i:= 1 to 5 do
```

```
     begin  write(chr(14));write(chr(2));  end;  textcolor(lightmagenta);highvideo;
     for i:= 1 to  5 do
     begin  write(chr(3));write(chr(11));   end; readln;
 end;
{****************************************************}
procedure   report;
const  max = 50;
type
      arr = array[1..max] of summaryrec;
var
      j,i,n,totvol:integer;  sum,tot:real;    a     :arr;
   begin
     clrscr;   totvol:=0; tot:=0.0;  sum:=0.0;  j:=0; assign(fp_s,f4);  reset(fp_s);
     while not  eof(fp_s) do
        begin   inc(j);  read(fp_s,a[j]);     end;
     close(fp_s);  textcolor(lightgreen+128);highvideo;  gotoxy(35,4);writeln(shopname);
     textcolor(blue+128);  highvideo; gotoxy(32,5);writeln(taxid);  textcolor(lightcyan);highvideo;
     gotoxy(2,7);
     for i:= 1 to  8 do
      begin write(chr(4)); write(chr(5)); end; textcolor(lightgreen);highvideo;
      for i:= 1 to 8 do
      begin write(chr(13)); write(chr(3)); end; textcolor(lightmagenta);highvideo;
      for i:= 1 to  8 do
      begin write(chr(4)); write(chr(14)); end; textcolor(lightgray);highvideo;
      for i:= 1 to 8 do
      begin write(chr(14)); write(chr(2));end; textcolor(lightmagenta);highvideo;
      for i:= 1 to  8 do
      begin write(chr(3)); write(chr(11)); end; textcolor(lightcyan);highvideo;
      for i:= 1 to  8 do
      begin write(chr(4)); write(chr(5)); end; textcolor(lightgreen);highvideo;
      for i:= 1 to 8 do
      begin write(chr(13));write(chr(3)); end; textcolor(lightmagenta);highvideo;
      for i:= 1 to  10 do
      begin write(chr(4)); write(chr(14)); end; textcolor(lightgray);highvideo;
      for i:= 1 to 9 do
      begin write(chr(14));write(chr(2)); end;textcolor(lightmagenta);highvideo;
      for i:= 1 to  8 do
```

```
  begin write(chr(3));  write(chr(11)); end; textcolor(lightgreen);highvideo;
  for i:= 1 to 8 do
  begin  write(chr(13));write(chr(3)); end;  textcolor(lightmagenta);highvideo;
  for i:= 1 to  8 do
  begin  write(chr(4)); write(chr(14)); end;textcolor(lightcyan);highvideo;
  for i:= 1 to  6 do
  begin write(chr(4)); write(chr(5)); end; textcolor(lightgreen);highvideo;
  for i:= 1 to 10 do
  begin  write(chr(13));write(chr(3)); end; textcolor(lightmagenta);highvideo;
  for i:= 1 to  10 do
  begin write(chr(4)); write(chr(14)); end; textcolor(lightred);highvideo;
 gotoxy(1,8);write('no'); gotoxy(12,8);write('id'); gotoxy(22,8);write('description');
 gotoxy(34,8);write('volumn'); gotoxy(44,8);write('profit'); gotoxy(57,8);write('total');
 n:=9; textcolor(lightgray +128);highvideo;  gotoxy(1,9);
 for i := 1 to 80 do     write(chr(3));     textcolor(lightcyan);highvideo;
  for i:= 1 to j do
   begin
       with a[i] do
        begin   inc(n);gotoxy(1,n);write(j);gotoxy(12,n);write(id);
                 gotoxy(22,n);write(name);      gotoxy(34,n);write(totalsell:2:2);
                 gotoxy(44,n);write(totalgoods:2);   gotoxy(57,n);write(profitperonce:2:2);
                 tot:=tot + totalsell;totvol:=totvol + totalgoods; sum:=sum + profitperonce;
        end;
   end;
textcolor(lightmagenta);highvideo;writeln;writeln; writeln(' total volumn = ',totvol);
writeln(' "  " profit = ',sum:2:2); writeln(' "  " summary = ',tot:2:2);
textcolor(red+128);highvideo;
 for i:= 1 to  5 do
  begin  write(chr(3));  write(chr(12));  end;   textcolor(lightgreen);highvideo;
  for i:= 1 to 5 do
  begin  write(chr(14));write(chr(2)); end;  textcolor(lightgreen);highvideo;
  for i:= 1 to 5 do
  begin write(chr(13)); write(chr(3));  end; textcolor(lightmagenta);highvideo;
  for i:= 1 to  5 do
  begin  write(chr(4));  write(chr(14)); end; textcolor(lightmagenta);highvideo;
  for i:= 1 to  5 do
  begin  write(chr(3)); write(chr(12)); end;textcolor(lightgray);highvideo;
```

```
        for i:= 1 to 5 do
        begin write(chr(14));   write(chr(2)); end;  textcolor(lightmagenta);highvideo;
        for i:= 1 to 5 do
        begin  write(chr(3));  write(chr(11));  end;textcolor(lightmagenta);highvideo;
        for i:= 1 to  5 do
        begin  write(chr(3));  write(chr(12));  end;  textcolor(lightgray);highvideo;
        for i:= 1 to 5 do
        begin  write(chr(14));  write(chr(2));  end; textcolor(lightmagenta);highvideo;
        for i:= 1 to 5 do
        begin  write(chr(3));write(chr(11));   end; textcolor(lightgreen);highvideo;
        for i:= 1 to 5 do
        begin  write(chr(14));  write(chr(2)); end;  readln;
end;
    {*****************************************************}
    procedure  checkstock;
    var   data : recmaster;    i,n:integer;
    begin
       clrscr;    n:=11; textcolor(blue+128);highvideo;   gotoxy(32,4);writeln(shopname);
       textcolor(yellow +128);highvideo;      gotoxy(29,5);writeln(taxid);
       textcolor(lightcyan);highvideo;           gotoxy(13,7);
     for i:= 1 to  5 do
        begin  write(chr(3));  write(chr(12)); end; textcolor(lightgreen);highvideo;
        for i:= 1 to 5 do
        begin   write(chr(14));  write(chr(2));  end;  textcolor(lightmagenta);highvideo;
        for i:= 1 to  5 do
        begin  write(chr(3));  write(chr(12)); end;  textcolor(lightgray);highvideo;
        for i:= 1 to 5 do
        begin  write(chr(14));  write(chr(2)); end;   textcolor(lightmagenta);highvideo;
        for i:= 1 to  5 do
        begin  write(chr(3)); write(chr(11)); end;  textcolor(lightmagenta);highvideo;
       assign(fp_t,f2);   reset(fp_t);   gotoxy(1,9);write('id'); gotoxy(7,9);write('description');
       gotoxy(23,9);write('onhand'); gotoxy(34,9);write('costprice');  gotoxy(50,9);write('sell price');
       gotoxy(64,9);writeln('day import'); textcolor(lightgreen+128);highvideo;       gotoxy(1,10);
       for i:= 1 to  20 do   write(chr(14));   textcolor(lightblue+128);highvideo;
       for i:= 1 to  20 do   write(chr(5));   textcolor(lightred+128);highvideo;
       for i:= 1 to  20 do  write(chr(3));   textcolor(yellow+128);highvideo;
       for i:= 1 to  20 do  write(chr(4));      textcolor(lightcyan);highvideo;
```

```
          while not eof(fp_t) do
          begin
            read(fp_t,data);     inc(n);
            with  data do
             begin
              gotoxy(1,n);write(id);  gotoxy(7,n);write(des);gotoxy(23,n);write(onhand);
              gotoxy(34,n);write(cost_price:2:2);   gotoxy(50,n);write(sell_price:2:2);
              gotoxy(64,n);write(day_import);
            end;
         end;
       writeln;writeln;      textcolor(lightred+128);        highvideo;
       for i:= 1 to 20 do   write(chr(2));  textcolor(lightgreen+128);     highvideo;
       for i:= 1 to 20 do  write(chr(3));    textcolor(lightcyan+128);     highvideo;
       for i:= 1 to 20 do   write(chr(6));  textcolor(lightmagenta+128);  highvideo;
       for i:= 1 to 20 do   write(chr(12));    close(fp_t);readln;
   end;
   begin
    repeat
      clrscr;   gotoxy(37,5); textcolor(yellow+128); highvideo;
     writeln('summary'); gotoxy(35,6); textcolor(red);
     writeln(shopname);  gotoxy(32,7); writeln(taxid);
     gotoxy(18,9); textcolor(green);
     for i:= 1 to 45  do  write(chr(5));   gotoxy(25,11); textcolor(lightcyan); highvideo;
     writeln('1 fide summary per day ');  gotoxy(25,12);
     writeln('2 report ');gotoxy(25,13);    writeln('3 check stock'); gotoxy(25,14);
     writeln('4 exit ');gotoxy(27,15);      textcolor(white);write(' enter :');
     readln(choice);     case choice of
      1:profit1;       2:report;       3:checkstock;     else
        begin
           if (choice <> 1 )  and (choice <> 2)   and   (choice <> 3)   and (choice <> 4)
           then  begin      gotoxy(29,17);writeln(' invalid please  enter  again');
        end;
        end;
       end;
       until  (choice  = 4);
end;
```

```
{**********************************************************}
procedure  deleteinsert;
var  choice:byte;
 { +++++++++++++++++  delete  ++++++++++++++++++++++}
    procedure       delete;
      var
          a:group_1; code:string[5];ind1:group_2; data1:indexrec;num,i,j:integer; test:boolean;
     begin
        clrscr; write('enter id to be delete = '); readln(code);  num:=0;
        assign(fp_i,f3);     reset(fp_i);
         while   not eof(fp_i) do
         begin   num:=num+1;    read(fp_i,data1);     ind1[num]:=data1;     end;
         close(fp_i);
         i:=1;       test:=true;
         while (i <= num) and test do
                        if code = ind1[i].id then   test:=false;    else    i:=i+1;
          if not test then       for j:= i to num - 1 do     ind1[j]:=ind1[j+1];
          assign(fp_i,f3);      rewrite(fp_i);
          for i:= 1 to num -1 do write(fp_i,ind1[i]);
          close(fp_i);
          num:=0;   assign(fp_m,f1);  reset(fp_m);
           while not eof(fp_m) do
                  begin   num:=num+1;  read(fp_m,a[num]);end;
          close(fp_m);    i:=1;          test:=true;
           while (i<=num) and test do
           if code =a[i].id then   test:=false    else    i:=i+1;
           if not test then
                     for j:= i to num-1 do       a[j]:=a[j+1];
          assign(fp_m,f1);  rewrite(fp_m);  for i:= 1 to num -1 do  write(fp_m,a[i]);
          close(fp_m);
      end;
   {+++++++++++++++++    insert ++++++++++++++++}
         { add data to file }
         procedure  insert;
         var
             data:recmaster;    data1:indexrec;    num,j,i:integer;   f:group_2;
             test:boolean;     temp:indexrec;
          begin
            clrscr;  assign(fp_m,f1);  reset(fp_m);     seek(fp_m,filesize(fp_m));
            with data do
            begin
               write('enter  id :');  readln(id);    write(' description  :');  readln(des);
               write('  onhand   :'); readln(onhand); write('  cost  price  :');
               readln(cost_price);  write('  sell price  :');  readln(sell_price);
               write('  day import  :');   readln(day_import);
             end;
            write(fp_m,data); close(fp_m);   num:=0;
           assign(fp_i,f3);  reset(fp_i);   seek(fp_i,filesize(fp_i));
          data1.id:=data.id;  data1.relative_address:=filepos(fp_i);   write(fp_i,data1);
          close(fp_i);    num:=0;   assign(fp_i,f3);   reset(fp_i);
            while  not eof(fp_i) do
             begin  num:=num+1;  read(fp_i,f[num]);    end;
            close(fp_i);  i:=1;     test:=true;
              while (i < num) and test do
               begin     test:=false;
                   for j:= 1 to num - i do
                     begin
                       if (f[j].id > f[j+1].id) then
                         begin data1:=f[j]; f[j]:=f[j+1];  f[j+1]:=data1;  test:=true;   end;
                   end;   inc(i);
              end;   j:=1;
         assign(fp_i,f3);   rewrite(fp_i);
         for i:= 1 to num do
          begin  write(fp_i,f[i]);      end;
        close(fp_i);
end;
  { ***************      edit dta ++++++++++++++++++}
  procedure  editdata;
  var  a:group_1;   num,i:integer;   test:boolean; data:recmaster;   code:string[5];
    begin
      clrscr;  num:=0;   assign(fp_m,f1);    reset(fp_m);
       while not eof(fp_m) do
         begin  num:=num+1;  read(fp_m,a[num]);    end;
       close(fp_m);
       write('enter code that edit data [end = *] :');      readln(code);
       while code <> '*' do
       begin      i:=1;    test:=true;
        while (i<=num) and test do
              if code = a[i].id then     test:=false      else     i:=i+1;
         if not test then
         begin    data.id:=code;
              with data do
               begin
                  write('enter  description :');  readln(des);
                  write('       onhand       :');  readln(onhand);
                 write('        cost price  :');    readln(cost_price);
                 write('       sell price   :');    readln(sell_price);
                write('enter day import      :');  readln(day_import);
              end;{end  with }      a[i]:=data;
       end
        else
             writeln('please try again');
          assign(fp_m,f1);        rewrite(fp_m);
          for i:= 1 to num do           write(fp_m,a[i]);
          close(fp_m);
          write('enter code that edit data [end = *] :');       readln(code);
   end;
end;
      {*****************************************************}
 begin
     repeat
     clrscr;textcolor(blue);highvideo;gotoxy(25,4);
     writeln(' 1    delete  record  data  that you want ');
    gotoxy(25,5); writeln(' 2    insert  data ');   gotoxy(25,6);writeln(' 3    edit data    ');
     gotoxy(25,7);writeln(' 4    exit ');            gotoxy(28,9);write('  choice -> ');readln(choice);
     case choice of
      1:delete;  2:insert;     3:editdata;
      else
      if (choice <> 1) and (choice <> 2) and  (choice <> 3) and (choice <> 4)
        then
               writeln(' invalid please enter again');
              end;
          until (choice = 4);
        end;
      {5555555555555555boonchok    555555555555555555555555555551
  begin
    clrscr;        textbackground(black);
    repeat
      dosdatetime;startsound;ticsound;      displaymenu(mainmenu);displayframe( 10.40);
      getmenuresponse(mainmenu,response);
    { menu choice }
    case response of
    '1':createfile;      '2':enterdata;      '3':readallfile;      '4':search;      '5':sellgood;
    '6':updatedata;      '7':summary;        '8':deleteinsert;     'e':exit;
     end;
    writeln;textbackground(0);clrscr;
   until response = 'e';
  readln
 end.
```