program bilet19;

uses crt;

type

uk = ^elem;

elem = record

num: integer;

tg: string;

name: string;

year: string;

price: string;

next, prev: uk;

end;

var first, last : uk;

F1, F2: text;

procedure p1(var first, last: uk);

var nov, p, u, before: uk;

str: string;

i, j: integer;

begin

new(u);

u^.prev := nil;

first := u;

new(u^.next);

u := u^.next;

u^.next := nil;

last := u;

while not eof(F1) do

begin

readln(F1, str);

writeln(F2, str);

new(nov);

i := 1;

nov^.num := 0;

while str[i] <> ' ' do

begin

nov^.num := nov^.num \* 10 + ord(str[i]) - 48;

i := i + 1;

end;

i := i + 1;

j := i;

while str[j] <> ' ' do

j := j + 1;

nov^.tg := copy(str, i, j - i);

i := j + 1;

j := i;

while str[j] <> ' ' do

j := j + 1;

nov^.name := copy(str, i, j- i);

i := j + 1;

j := i;

while str[j] <> ' ' do

j := j +1;

nov^.year := copy(str, i, j - i);

i := j +1;

nov^.price := copy(str, i, length(str) - i + 1);

p := first;

while (p^.next^.next <> nil) and (p^.next^.num < nov^.num) do

p := p^.next;

nov^.next := p^.next;

p^.next := nov;

end;

u := first^.next;

before := first;

while u <> nil do

begin

u^.prev := before;

before := u;

u := u^.next;

end;

end;

procedure p2(var first, last: uk);

var nov, sled, cur: uk;

str: string;

i, j: integer;

begin

writeln('Enter new:');

readln(str); {

new(nov);

i := 1;

nov^.num := 0;

while str[i] <> ' ' do

begin

nov^.num := nov^.num \* 10 + ord(str[i]) - 48;

i := i + 1;

end;

i := i + 1;

j := i;

while str[i] <> ' ' do

j := j +1;

nov^.tg := copy(str, i, j - i);

i := j + 1;

nov^.other := copy(str, i, length(str) - i +1); }

new(nov);

i := 1;

nov^.num := 0;

while str[i] <> ' ' do

begin

nov^.num := nov^.num \* 10 + ord(str[i]) - 48;

i := i + 1;

end;

i := i + 1;

j := i;

while str[j] <> ' ' do

j := j + 1;

nov^.tg := copy(str, i, j - i);

i := j + 1;

j := i;

while str[j] <> ' ' do

j := j + 1;

nov^.name := copy(str, i, j- i);

i := j + 1;

j := i;

while str[j] <> ' ' do

j := j +1;

nov^.year := copy(str, i, j - i);

i := j +1;

nov^.price := copy(str, i, length(str) - i + 1);

cur := first;

while (cur^.next^.next <> nil) and (cur^.next^.num < nov^.num) do

cur := cur^.next; {

nov^.next := cur^.next;

nov^.prev := cur;

cur^.next := nov;

cur := nov^.next;

cur^.prev := nov; }

sled := cur^.next;

sled^.prev := cur;

nov^.next := sled;

sled^.prev := nov;

cur^.next := nov;

nov^.prev := cur;

end;

procedure p3(var first, last: uk);

var sach, tg: string;

p : uk;

begin

writeln('Enter name book');

readln(sach);

writeln('Enter author');

readln(tg);

writeln(F2);

writeln(F2,'=========================');

if (tg > 'A') and (tg < 'KZ') then

begin

p := first^.next;

while (p^.next <> nil) and (p^.tg <> tg) do

p := p^.next;

writeln(F2,'Name book: ', p^.name, 'Author: ', p^.tg);

writeln(F2,'Writed in ', p^.year, 'Price: ', p^.price, ' rup.');

end

else

begin

p := last^.prev;

while (p^.prev <> nil) and (p^.tg <> tg) do

p := p^.prev;

writeln(F2,'Name book: ', p^.name, 'Author: ', p^.tg);

writeln(F2,'Writed in ', p^.year, 'Price: ', p^.price, ' rup.');

end;

end;

procedure p4(var first, last: uk);

var cur : uk;

begin

writeln(F2);

writeln(F2,'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln(F2);

cur := first^.next;

while cur^.next <> nil do

begin

writeln(F2, cur^.num,' ', cur^.tg,' ',cur^.name,' ', cur^.year,' ', cur^.price);

cur := cur^.next;

end;

writeln(F2);

cur := last^.prev;

while cur^.prev <> nil do

begin

writeln(F2, cur^.num, ' ', cur^.tg,' ', cur^.name,' ', cur^.year,' ', cur^.price);

cur := cur^.prev;

end;

end;

BEGIN

clrscr;

assign(F1, 'f19.inp');

reset(F1);

assign(F2, 'f19.out');

rewrite(F2);

p1(first, last);

p4(first, last);

p2(first, last);

p4(first, last);

p3(first, last);

p4(first, last);

close(F1);

close(F2);

end.