program lab3\_1;

uses

Crt;

type

PInf = ^TInf;

TInf = record { ‡ ЇЁбм ¤«п н«Ґ¬Ґ­в  ®зҐаҐ¤Ё }

num: Integer; { —Ёб«® }

next: PInf; { “Є § вҐ«м ­  б«Ґ¤гойЁ© н«Ґ¬Ґ­в ®зҐаҐ¤Ё }

end;

var

c, m: Longint; { Љ®«ЁзҐбвў® ®ЇҐа жЁ© ба ў­Ґ­Ёп Ё ЇҐаҐбл«®Є }

{ Џа®ўҐапҐв ®зҐаҐ¤м ­  Їгбв®вг }

function IsQueueEmpty(qHead, qTail: PInf): Boolean;

begin

IsQueueEmpty := qHead = nil;

end;

{ “бв ­ ў«Ёў Ґв б«Ґ¤гойЁ© н«Ґ¬Ґ­в ®зҐаҐ¤Ё }

procedure SetQueueNext(var qHead, qTail: PInf; next: PInf);

begin

if IsQueueEmpty(qHead, qTail) then

qHead := next

else

qTail^.next := next;

qTail := next;

end;

{ „®Ў ў«пҐв ­®ўл© н«Ґ¬Ґ­в ў ®зҐаҐ¤м }

procedure AddQueueNext(var qHead, qTail: PInf; num: Integer);

var

p: PInf;

begin

New(p);

p^.num := num;

p^.next := nil;

SetQueueNext(qHead, qTail, p);

end;

{ „Ґ« Ґв ®зҐаҐ¤м Їгбв®© }

procedure EmptyQueue(var qHead, qTail: PInf);

begin

qHead := nil;

qTail := nil;

end;

{ ‘®авЁагҐв ®зҐаҐ¤м ¬Ґв®¤®¬ Їап¬®Ј® б«Ёп­Ёп }

procedure Sort(var qHead, qTail: PInf);

var

aHead: array[0..1] of PInf; { “Є § вҐ«Ё ­  ­ з «  а Ў®зЁе ®зҐаҐ¤Ґ© }

aTail: array[0..1] of PInf; { “Є § вҐ«Ё ­  Є®­жл а Ў®зЁе ®зҐаҐ¤Ґ© }

i, k: Integer;

cHead: array[0..1] of PInf; { “Є § вҐ«Ё ­  ­ з «  ®зҐаҐ¤Ґ© ¤«п б«Ёп­Ёп }

cTail: array[0..1] of PInf; { “Є § вҐ«Ё ­  Є®­жл ®зҐаҐ¤Ґ© ¤«п б«Ёп­Ёп }

p: Integer; { ЏаҐ¤Ї®« Ј Ґ¬л© а §¬Ґа бҐаЁЁ }

qr: array[0..1] of Integer; { ђ §¬Ґал бҐаЁ© ¤«п а Ў®зЁе ®зҐаҐ¤Ґ© }

\_p: PInf;

n: Integer; { Љ®«ЁзҐбвў® н«Ґ¬Ґ­в®ў ў ®зҐаҐ¤Ё }

begin

c := 0;

m := 0;

for i := 0 to 1 do

EmptyQueue(aHead[i], aTail[i]);

n := 0;

k := 0;

\_p := qHead;

while \_p <> nil do begin { „Ґ« Ґ¬ а бйҐЇ«Ґ­ЁҐ ®зҐаҐ¤Ё ­  2 ®зҐаҐ¤Ё }

SetQueueNext(aHead[k], aTail[k], \_p);

Inc(m);

Inc(n);

k := 1 - k; { ЊҐ­пҐ¬ ®зҐаҐ¤м ­  ¤агЈго }

\_p := \_p^.next;

end;

for k := 0 to 1 do

aTail[k]^.next := nil;

p := 1; { Ќ зЁ­ Ґ¬ ®б­®ў­®©  «Ј®аЁв¬ б®авЁа®ўЄЁ }

while p < n do begin

for k := 0 to 1 do

EmptyQueue(cHead[k], cTail[k]);

i := 0;

{ Џ®Є  ў а Ў®зЁе ®зҐаҐ¤пе Ґбвм н«Ґ¬Ґ­вл }

while (aHead[0] <> nil) or (aHead[1] <> nil) do begin

for k := 0 to 1 do begin

qr[k] := 0;

if aHead[k] <> nil then

qr[k] := p;

end;

{ ђҐ «Ё§®ўлў Ґ¬  «Ј®аЁв¬ б«Ёп­Ёп }

while (qr[0] > 0) and (qr[1] > 0) do begin

case aHead[0]^.num < aHead[1]^.num of

True: k := 0;

False: k := 1;

end;

Inc(c);

SetQueueNext(cHead[i], cTail[i], aHead[k]);

Inc(m);

{ ЏҐаҐ¬Ґй Ґ¬ гЄ § вҐ«м ­ з «  а Ў®зҐ© ®зҐаҐ¤Ё ўЇҐаҐ¤ }

aHead[k] := aHead[k]^.next;

if aHead[k] <> nil then

Dec(qr[k])

else

qr[k] := 0;

end;

k := -1;

if qr[0] > 0 then { …б«Ё ў а Ў®зҐ© ®зҐаҐ¤Ё 0 ҐйҐ ®бв «Ёбм н«Ґ¬Ґ­вл }

k := 0

else if qr[1] > 0 then { …б«Ё ў а Ў®зҐ© ®зҐаҐ¤Ё 0 ҐйҐ ®бв «Ёбм н«Ґ¬Ґ­вл }

k := 1;

if k in [0, 1] then

while (qr[k] > 0) and (aHead[k] <> nil) do begin

SetQueueNext(cHead[i], cTail[i], aHead[k]);

Inc(m);

aHead[k] := aHead[k]^.next;

Dec(qr[k]);

end;

i := 1 - i;

end;

for k := 0 to 1 do

cTail[k]^.next := nil;

for k := 0 to 1 do

aHead[k] := cHead[k]; { Џ®«гз Ґ¬ ­®ўлҐ а Ў®зЁҐ ®зҐаҐ¤Ё }

p := 2 \* p; { “ўҐ«ЁзЁў Ґ¬ а §¬Ґа бҐаЁЁ }

end;

qHead := cHead[0];

qTail := cTail[0];

end;

{ ‡ Ї®«­пҐв ®зҐаҐ¤м б«гз ©­л¬Ё зЁб« ¬Ё ®в 0 ¤® 99 }

procedure RandomQueue(var qHead, qTail: PInf);

var

n, i: Integer;

begin

Write('‚ўҐ¤ЁвҐ Є®«ЁзҐбвў® н«Ґ¬Ґ­в®ў ў Ї®б«Ґ¤®ў вҐ«м­®бвЁ: ');

Readln(n);

Writeln;

EmptyQueue(qHead, qTail);

Randomize;

for i := 1 to n do

AddQueueNext(qHead, qTail, Random(100));

end;

{ ‚лў®¤Ёв ®зҐаҐ¤м ­  нЄа ­ }

procedure PrintQueue(qHead, qTail: PInf);

var

p: PInf;

begin

p := qHead;

while p <> nil do begin

Write(p^.num, ' ');

p := p^.next;

end;

Writeln;

end;

{ ‚лў®¤Ёв ­  нЄа ­ Є®«ЁзҐбвў® ®ЇҐа жЁ© ба ў­Ґ­Ёп Ё ЇҐаҐбл«®Є }

procedure PrintInf;

begin

Writeln('C = ', c, ', M = ', m);

end;

{ ‚лў®¤Ёв ­  нЄа ­ Ё­д®а¬ жЁо ®Ў ®зҐаҐ¤Ё }

procedure Print(var qHead, qTail: PInf; s: String);

begin

Writeln(s, ':');

PrintQueue(qHead, qTail);

Writeln;

Sort(qHead, qTail);

Writeln('Џ®б«Ґ¤®ў вҐ«м­®бвм Ї®б«Ґ б®авЁа®ўЄЁ:');

PrintQueue(qHead, qTail);

Writeln;

PrintInf;

ReadKey;

end;

var

qHead, qTail: PInf; { “Є § вҐ«Ё ­  ­ з «® Ё Є®­Ґж ®зҐаҐ¤Ё }

begin

ClrScr;

Writeln('ЊҐв®¤ Їап¬®Ј® б«Ёп­Ёп');

Writeln;

RandomQueue(qHead, qTail);

Print(qHead, qTail, '‘«гз ©­ п Ї®б«Ґ¤®ў вҐ«м­®бвм');

Writeln;

Print(qHead, qTail, '“Ї®ап¤®зҐ­­ п Ї®б«Ґ¤®ў вҐ«м­®бвм');

end.