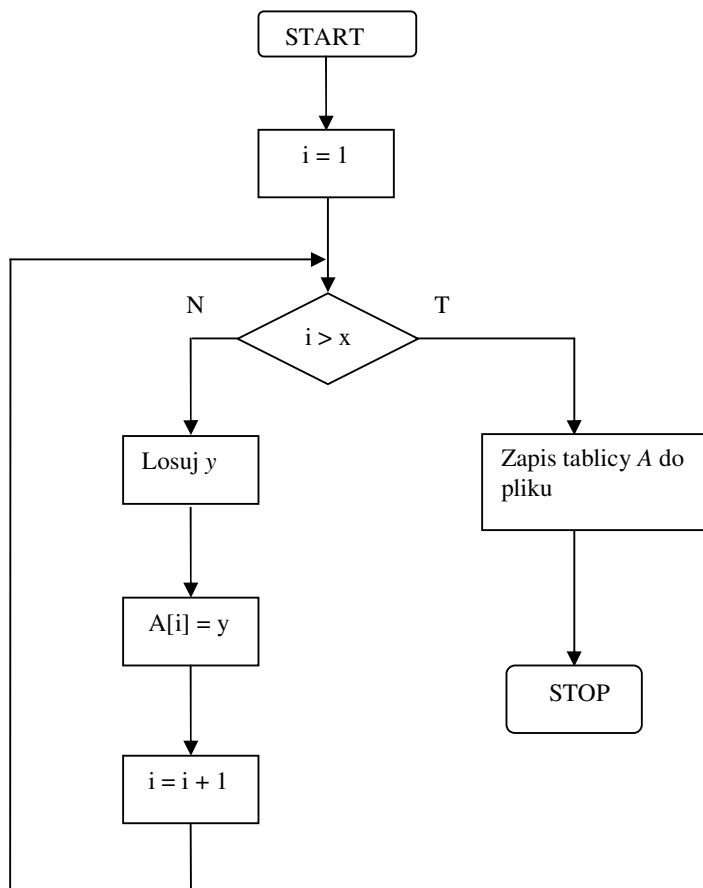


<b>Uniwersytet Zielonogórski</b>	Wykonali:	Grupa:	Nr ćwiczenia: 4	Ocena:
<b>Laboratorium</b>				
Temat ćwiczenia: Wybrane metody sortowania zewnętrzne.		Prowadzący:	Data wyk. ćw.	Data odd. spr.

1. Program losujący  $x$  elementową tablicę liczb typu integer i zapisujący wylosowaną tablicę  $A$  do pliku.

- schemat blokowy

kod źródłowy w języku Turbo Pascal



```
program wpis;
uses crt;
const x=7000;
type Tab = array[1..x] of integer;
var A:Tab; i:word; f:file of Tab; S:string;

BEGIN

Randomize;
S:='d:\zad1';

Assign(f,S);
Rewrite(f);

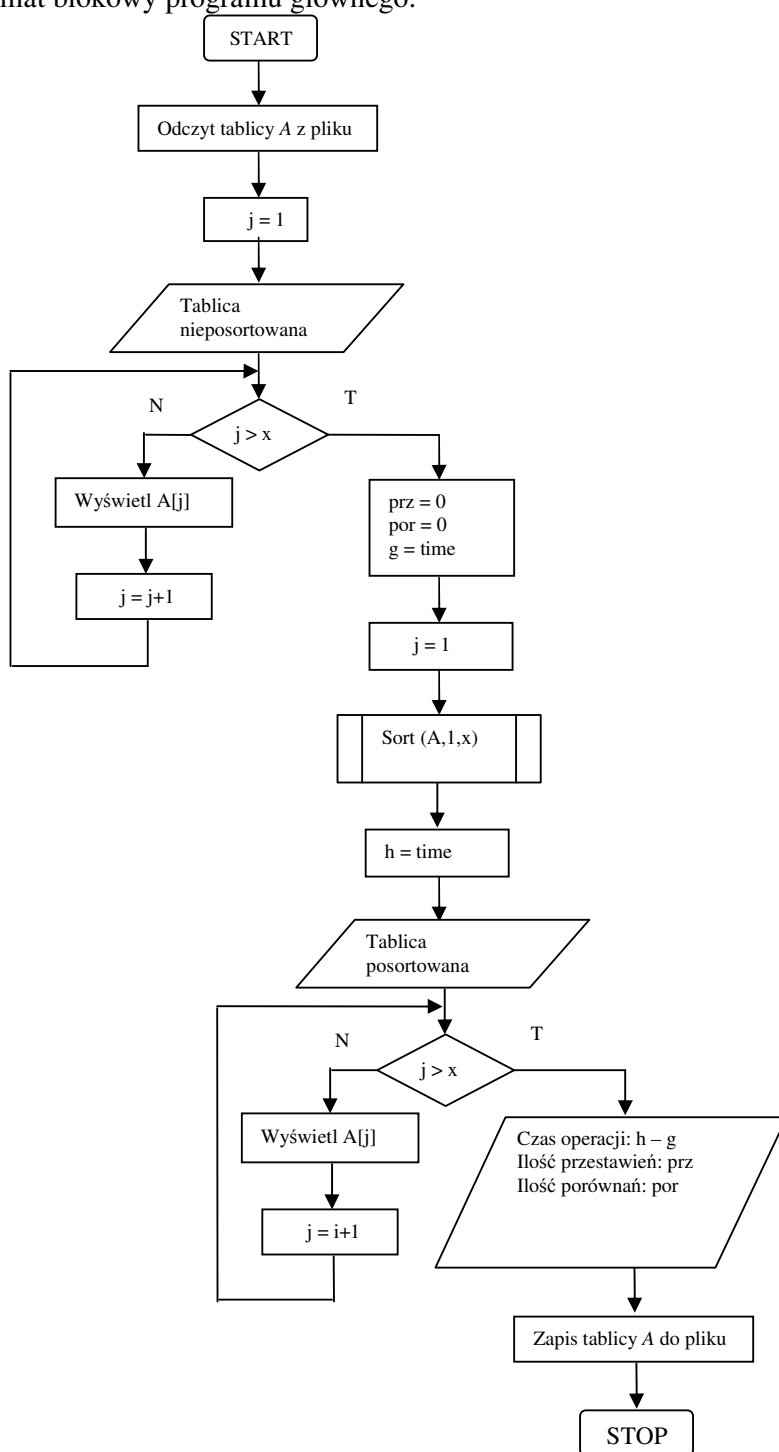
for i:=1 to x do
A[i]:=Random(3000)+1;

write(f,A);
Close(f);

END.
```

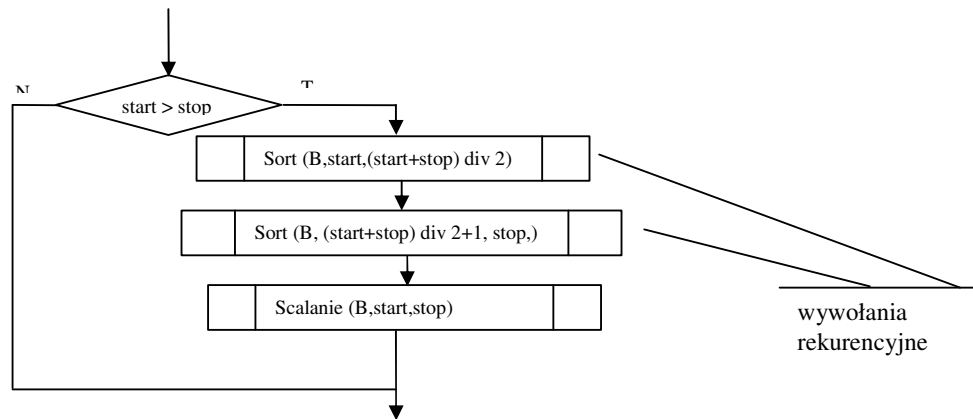
## 2. Sortowanie przez scalanie.

- schemat blokowy programu głównego:

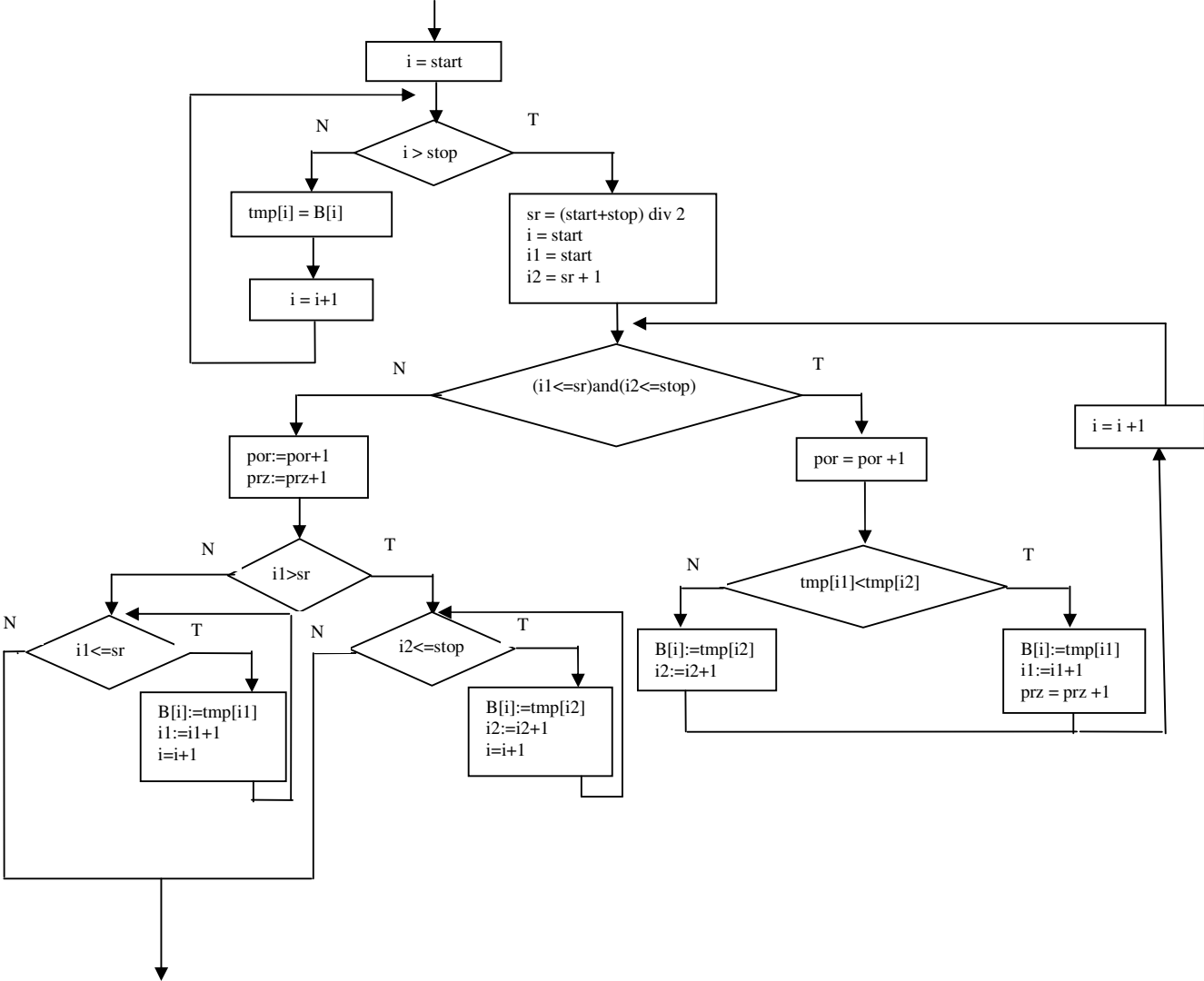


- schemat blokowy podprogramów:

a) procedura Sort(B,start, stop):



b) procedura Scalanie(B, strat, stop);



- kod źródłowy w języku Turbo Pascal

```

Program Scal;
uses crt,dos;
const x=7000;
Type Tab = array[1..x] Of integer;
var A:Tab; S,Sa:string; f,f1:file of Tab; g,m,s1,ss1,s2,ss2,r,j:word; prz,por:Longint;

```

```

Procedure Scalenie(var B:Tab; start,stop:integer);
var i,i1,i2,sr:Integer; tmp:Tab;
begin

```

```

for i:=start to stop do tmp[i]:=B[i];

```

```

sr:=(start+stop) div 2;

```

```

i:=start;

```

```

i1:=start;

```

```

i2:=sr+1;

```

```

while (i1<=sr)and(i2<=stop) do begin

```

```

  por:=por+1;

```

```

  If tmp[i1]<tmp[i2] then begin

```

```

    B[i]:=tmp[i1];

```

```

    prz:=prz+1;

```

```

    i1:=i1+1; end else begin

```

```

    B[i]:=tmp[i2];

```

```

    i2:=i2+1; end;

```

```

    i:=i+1;          end;

```

```

  por:=por+1;

```

```

  prz:=prz+1;

```

```

If i1>sr then

```

```

  while i2<=stop do begin

```

```

    B[i]:=tmp[i2];

```

```

    i2:=i2+1;

```

```

    i:=i+1; end else

```

```

  while i1<=sr do begin

```

```

    B[i]:=tmp[i1];

```

```

    i1:=i1+1;

```

```

    i:=i+1; end;

```

```

end;

```

```

Procedure Sort(var B:Tab; start,stop:integer);

```

```

begin

```

```

  If start<stop then begin

```

```

    Sort(B, start, (start+stop) div 2);

```

```

    Sort(B, ((start+stop) div 2)+1, stop);

```

```

    Scalenie(B, start, stop); end;

```

```

end;

```

```
BEGIN
Clrscr;
Randomize;

S:='d:\zad1';
Assign(f,S);
Reset(f);
Read(f,A);
Close(f);

writeln('TABLICA przed posortowaniem: ');
for j:=1 to x do write(A[j],' ');
por:=0;
prz:=0;

GetTime(g,m,s1,ss1);

Sort(A,1,x);

GetTime(g,m,s2,ss2);

writeln('TABLICA po posortowaniu: ');
for j:=1 to x do write(A[j],' ');

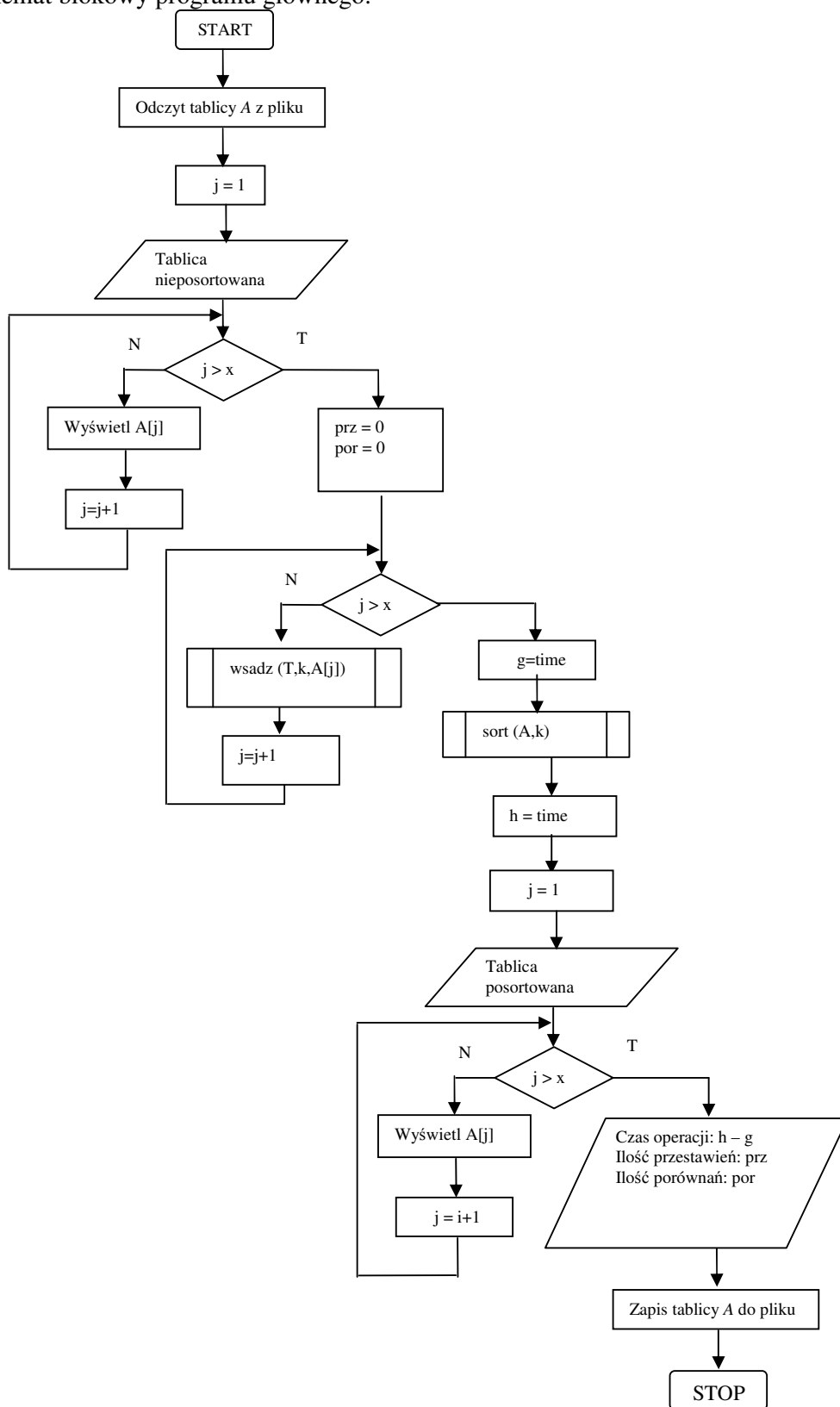
writeln('Czas rozpoczecia operacji: ',s1,' : ',ss1);
writeln('Czas zakonczenia operacji: ',s2,' : ',ss2);
writeln('Ilosc przestawien: ',prz);
writeln('Ilosc porownan: ',por);

Sa:='d:\zad_1';
Assign(f1,Sa);
Rewrite(f1);
write(f1,A);
Close(f1);

readln
END.
```

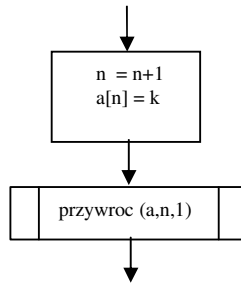
### 3. Sortowanie stogowe

- . schemat blokowy programu głównego:

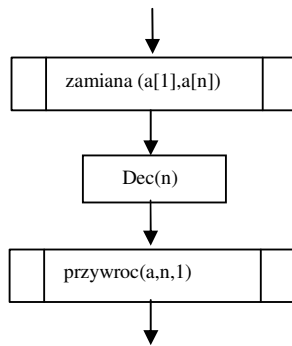


●Schematy blokowe podprogramów

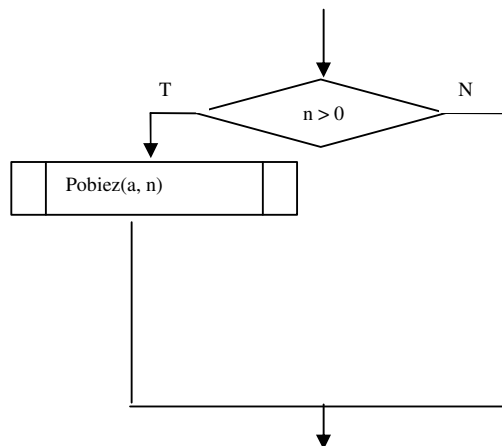
a)Procedura Wsadz(A,n,k)



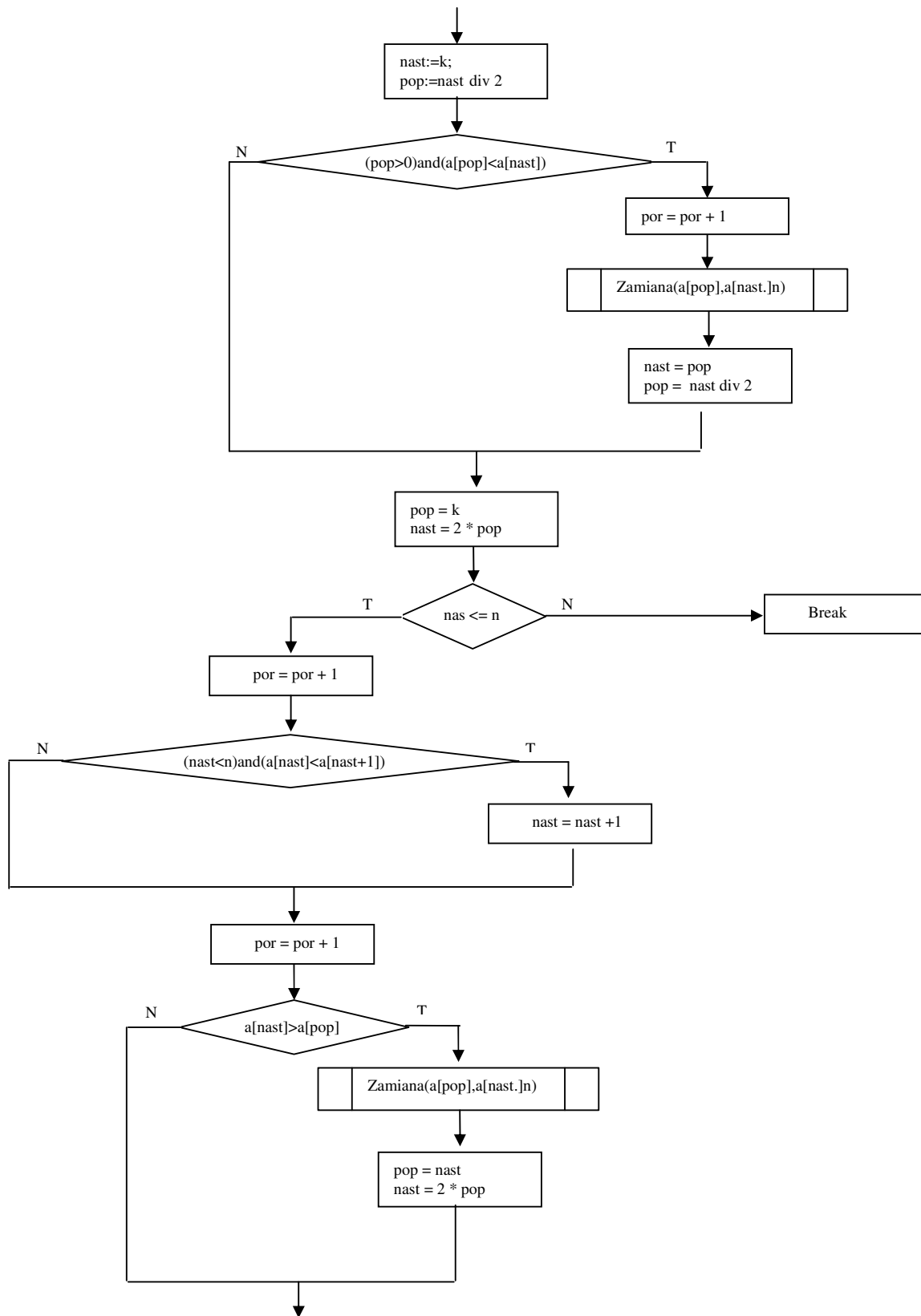
b)Procedura Pobierz(A,n)



c)Procedura Sort(A,n)



d) Procedura Przywroc (A,n,k)



- kod źródłowy w języku Turbo Pascal

```

program stog;
uses crt,dos;
const x=7000;
Type Tab=array[1..x] of integer;
Var A,T:Tab; S,Sa:string; f,f1:file of Tab; g,m,s1,ss1,s2,ss2,j,k:word;
    prz,por:Longint;

```

```

Procedure Zamiana(var a,b:integer);
var tmp:integer;
begin
prz:=prz+1;
tmp:=a;
a:=b;
b:=tmp;
end;

```

```

Procedure Przywroc(var A:Tab; n, k:word);
var pop,nast:word;
begin
nast:=k;
pop:=nast div 2;

```

```

    while (pop>0)and(a[pop]<a[nast]) do begin
        por:=por+1;
        Zamiana(a[pop], a[nast]);
        nast:=pop;
        pop:=nast div 2; end;

```

```

pop:=k;
nast:=2*pop;

```

```

    while nast<=n do begin
        por:=por+1;
        If (nast<n)and(a[nast]<a[nast+1]) then Inc(nast);
        por:=por+1;
        If a[nast]>a[pop] then begin
            Zamiana(a[pop], a[nast]);
            pop:=nast;
            nast:=2*pop; end else Break;
        end;

```

```

end;

```

```

Procedure Wsadz(var A:Tab; var n:word; k:word);
begin
Inc(n);
a[n]:=k;
Przywroc(a, n,k);
end;

```

```

Procedure Pobiez(var A:Tab; var n:word);
begin
  Zamiana(a[1], a[n]);
  Dec(n);
  Przywroc(a, n, 1);
end;

Procedure Sort(var A:Tab; var n:word);
begin
  while n>0 do Pobiez(a, n);
end;

Begin
Clrscr;
Randomize;
k:=0;
S:='d:\zad1';
Assign(f,S);
Reset(f);
Read(f,A);
Close(f);
writeln('TABLICA przed posortowaniem: ');
writeln;

for j:=1 to x do write(A[j], ' ');
por:=0;
prz:=0;
GetTime(g,m,s1,ss1);
for j:=1 to x do Wsadz(T, k,A[j] );
Sort(a, k);
GetTime(g,m,s2,ss2);
writeln; writeln; writeln;
writeln('TABLICA po posortowaniu: ');
writeln;
for j:=1 to x do write(A[j], ' ');
writeln; writeln;
writeln('Czas rozpoczecia operacji: ',s1,' : ',ss1);
writeln('Czas zakonczenia operacji: ',s2,' : ',ss2);
writeln('Ilosc przestawien: ',prz);
writeln('Ilosc porownan: ',por);
Sa:='d:\zad_2';
Assign(f1,Sa);
Rewrite(f1);
write(f1,A);
Close(f1);
readln;
End.

```

#### 4. Porównanie metod sortowania.

Metoda	Rozmiar tablicy	Ilość przestawień	Ilość porównań	Czas operacji [s]
Sortowanie przez scalanie	7 000			
Sortowanie stogowe	7 000			

#### **Wnioski:**

Nasze badania przeprowadzaliśmy na tablicach 7000 elementowych. Tablice były wypełniane liczbami z zakresu od 0 do 3000.