

## Bonus:

### Príklad č. 23: Algoritmus pre usporiadanie prvkov poľa metódou „prebublávania“ s kontrolou výmeny prvkov

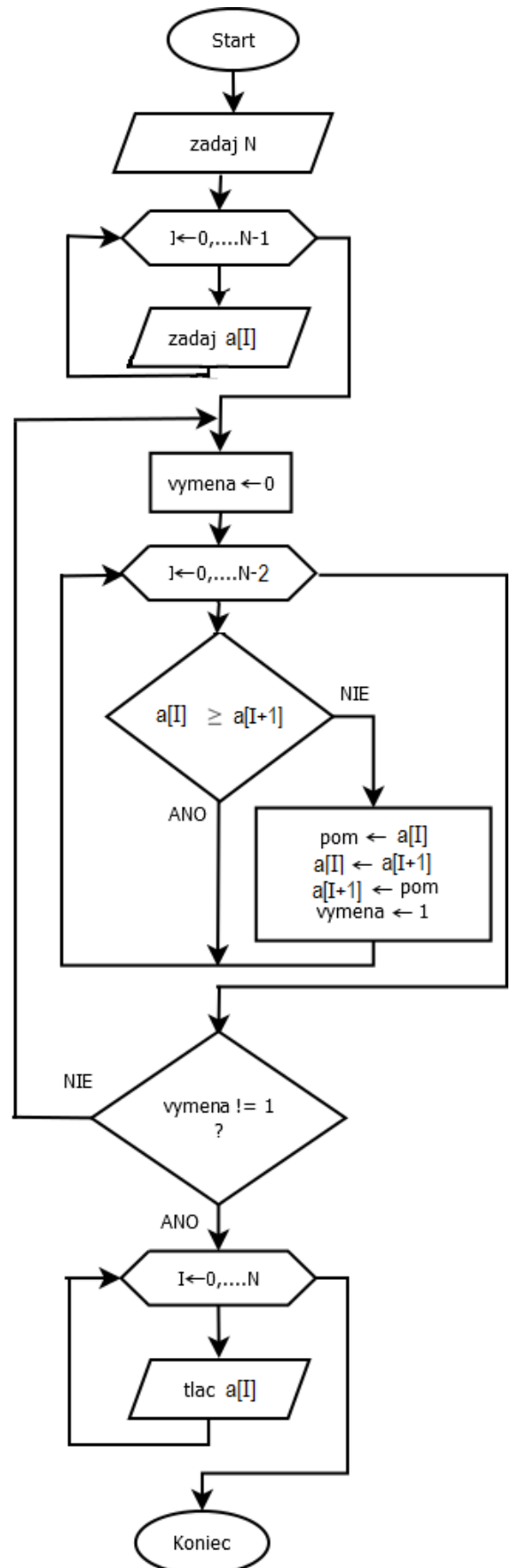
Vstupné premenné: jednorozmerné pole čísel A veľkosti N,  
Výstupné premenné: usporiadane pole čísel A  
Pomocné premenné: I,POM,VYMENA

#### Analýza riešenia :

Na vstupe zadáme jednorozmerné pole čísel A veľkosti N. V cykle sa postupne porovnávajú prvky  $A[J]$  a  $A[J+1]$ . Na začiatku cyklu sa príznak VYMENA, ktorý indikuje že nastala výmena počas konania cyklu, nastaví na hodnotu 0. Ak je hodnota  $A[J]$  väčšia ako hodnota  $A[J+1]$ , tak sa vymenia a  $VYMENA=1$ . Tento cyklus sa opakuje kým nastáva výmena prvkov, teda ak po skončení cyklu je hodnota príznaku VYMENA 1.

#### Postup (podľa knihy):

1. krok: Načítaj N
2. krok:  $I=1$
3. krok: Ak neplatí že  $I < N$  potom prejsť na krok 7.
4. krok: Čítaj  $A[I]$
5. krok:  $I=I+1$
6. krok: Prejsť na krok 3.
7. krok:  $VYMENA=0$
8. krok:  $I=1$
9. krok: Ak neplatí že  $I < N-1$  potom prejsť na krok 13.
10. krok: Ak  $A[I] > A[I+1]$  vymeniť navzájom hodnoty  $A[I]$  a  $A[I+1]$  a  $VYMENA = 1$
11. krok:  $I=I+1$
12. krok: Prejsť na krok 9.
13. krok: Ak  $VYMENA = 1$  potom prejsť na krok 7.
14. krok:  $I=1$
15. krok: Ak neplatí že  $I < N$  potom prejsť na krok 20.
16. krok: Výpis  $A[I]$
17. krok:  $I=I+1$
18. krok: Prejsť na krok 17.
19. krok: Koniec



## Prepis algoritmu v jazyku C:

```
#include<stdio.h>
#include<conio.h>

int main() {
    int N, I, J, pom, vymena;

    printf("Zadaj pocet cisel:");
    scanf("%d", &N);
    int a[N];

    printf("Zadaj %d prvkov vektora:", N);
    for (I=0; I<N; I++) {
        scanf("%d", &a[I]);
    }

    do{
        vymena=0;
        for (I=0; I<N-1; I++) {
            if (a[I]<a[I+1]) {
                pom=a[I];
                a[I]=a[I+1];
                a[I+1]=pom;
                vymena=1;
            }
        }
    }while (vymena==1);

    for (I=0; I<N; I++) {
        printf("%d, ", a[I]);
    }

    getch();
    return 0;
}
```

## Priebeh algoritmu v jazyku C:

```
-----Pred spracovaným-----  
Vektor A : 1, 2, 3, 4, 5, 6, 7, 8, 9,  
hodnota premennej vymena = 0  
  
-----PREBUBLAVANIE-----  
1.) vektor po prebublani : 2, 3, 4, 5, 6, 7, 8, 9, 1,  
hodnota premennej vymena = 1  
2.) vektor po prebublani : 3, 4, 5, 6, 7, 8, 9, 2, 1,  
hodnota premennej vymena = 1  
3.) vektor po prebublani : 4, 5, 6, 7, 8, 9, 3, 2, 1,  
hodnota premennej vymena = 1  
4.) vektor po prebublani : 5, 6, 7, 8, 9, 4, 3, 2, 1,  
hodnota premennej vymena = 1  
5.) vektor po prebublani : 6, 7, 8, 9, 5, 4, 3, 2, 1,  
hodnota premennej vymena = 1  
6.) vektor po prebublani : 7, 8, 9, 6, 5, 4, 3, 2, 1,  
hodnota premennej vymena = 1  
7.) vektor po prebublani : 8, 9, 7, 6, 5, 4, 3, 2, 1,  
hodnota premennej vymena = 1  
8.) vektor po prebublani : 9, 8, 7, 6, 5, 4, 3, 2, 1,  
hodnota premennej vymena = 1  
9.) vektor po prebublani : 9, 8, 7, 6, 5, 4, 3, 2, 1,  
hodnota premennej vymena = 0  
  
-----VEKTOR po spracovaní-----  
Vektor A:9, 8, 7, 6, 5, 4, 3, 2, 1, _
```