

Domáca úloha:

Príklad č. 22: Algoritmus pre usporiadanie prvkov poľa metódou „prebublávania“

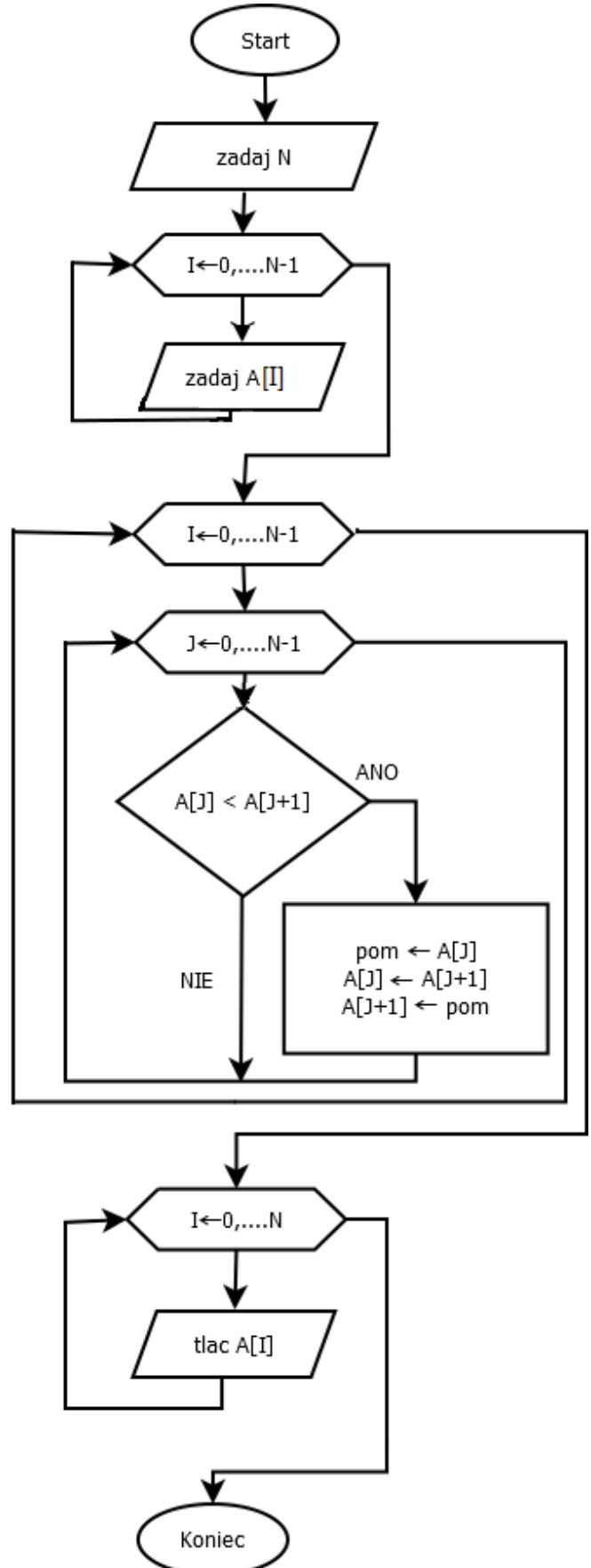
Vstupné premenné: jednorozmerné pole čísel A
veľkosti N,
Výstupné premenné: usporiadane pole čísel A
Pomocné premenné: I,J,pom

Analýza riešenia :

Na vstupe zadáme jednorozmerné pole čísel A veľkosti N. V cykle sa postupne porovnávajú prvky $A[J]$ a $A[J+1]$. Ak je hodnota $A[J]$ väčšia ako hodnota $A[J+1]$, tak sa vymenia. Tento cyklus je potrebné vykonať N-1 krát pre prípad že najmenší prvok sa nachádza na konci poľa.

Slovný popis algoritmu:

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



Prepis algoritmu v jazyku C:

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

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

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

    printf("Zadaj %d prvkov vektora:",N);
    for (I=1;I<=N;I++) {
        scanf ("%d", &a[I]);
    }

    for (I=1;I<N;I++) {
        for (J=1;J<N;J++) {
            if (a[J]<a[J+1]) {
                pom=a [J];
                a [J]=a [J+1];
                a [J+1]=pom;}
        }

    for (I=1;I<=N;I++) {
        printf ("%d, ",a[I]);
    }

    getch();
    return 0;
}
```