

Príklad č.28.: Vytvorte algoritmus, ktorý načíta čísla zo súboru „vstup.txt“, všetky čísla uloží do vektora A a každý druhý prvok zvýši o 1, teda zvýši hodnotu prvku na 2., 4., 6., ... mieste. Výsledný vektor sa uloží do súboru „vystup.txt“. Program na monitor vypíše hlásenie o načítaní čísel zo súboru a o uložení vektora do súboru.

Vstupné premenné: f - smerník na súbor (vstup.txt)

Pomocné premenné: pole[50] - pole na načítanie prvkov zo súboru

Systémové premenné: N - premenná indexu pre pole[50]

i - premenná cyklu

x - premenná indexu pre pole A[x]

Výstupná premenná: A[50] - výstupné pole so zmenenými prvkami

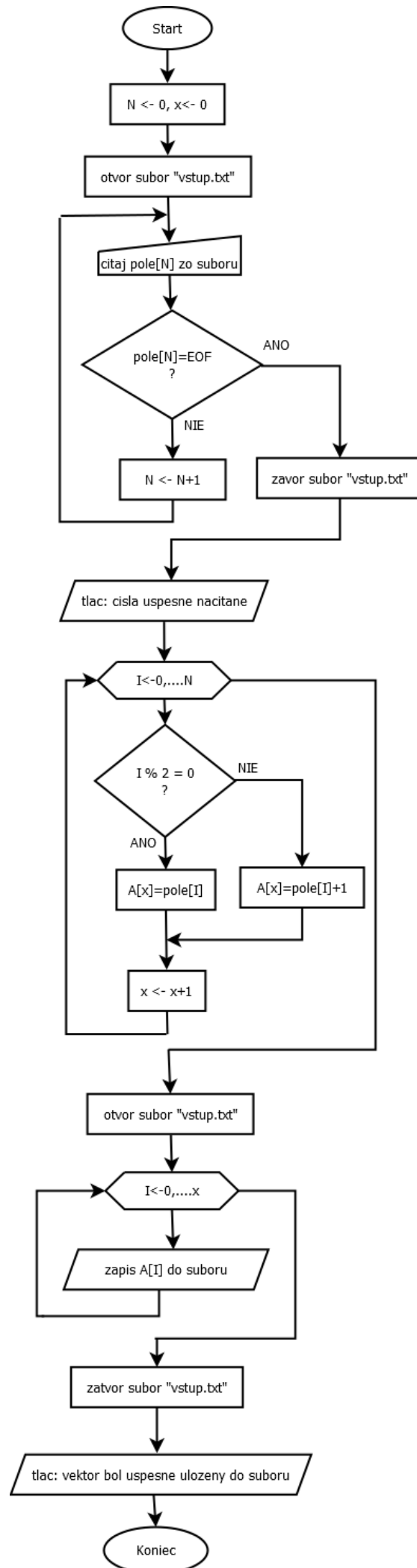
Metóda:

Je potrebné vytvoriť súbor „vstup.txt“, do ktorého zapíšeme ľubovoľný počet čísel oddelených medzerou. Po načítaní vstupného súboru získame pole čísel. Keďže číslo na 2., 4., 6., ... mieste má index 1, 3, 5 atď., budeme zvyšovať hodnotu čísel, ktorých index po delení dvojkou nedáva zvyšok 0. Výsledné pole potom uložíme do súboru „vystup.txt“.

Slovný popis algoritmu:

1. krok: priradiť $n \leftarrow 0$, $x \leftarrow 0$
2. krok: otvor súbor vstup.txt
3. krok: súbor neexistuje? ak neplatí pokračuj krokom 4, inak prejdí na krok 17
4. krok: čítaj pole[N] zo súboru
5. krok: nie je pole[N] na konci súboru? ak nie prejdí na krok 6, inak - krok 7
6. krok: $N++$, vráť sa na krok 4
7. krok: zatvor súbor vstup.txt
8. krok: ak i má hodnotu od 0 po N pokračuj krokom 9, inak prejdí na krok 13
9. krok: ak $i \% 2 = 0$ prejdí na krok 10, inak prejdí na krok 11
10. krok: $A[x] \leftarrow \text{pole}[i]$
11. krok: $A[x] \leftarrow \text{pole}[i] + 1$
12. krok: $x++$, vráť sa na krok 8
13. krok: otvor súbor vystup.txt
14. krok: ak i má hodnotu od 0 po x, pokračuj krokom 15, inak prejdí na krok 16
15. krok: zapíš $A[x]$ do súboru
16. krok: zatvor súbor vystup.txt
17. krok: Koniec

Vývojový diagram:



Prepis algoritmu v jazyku C:

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

int main() {

    FILE*f;
    int N,I,x,y;
    int pole[50];
    int a[50];

    N=0;
    x=0;

    f=fopen("vstup.txt","r");
    while (fscanf(f,"%d",&pole[N]) !=EOF)N++;
    fclose(f);

    printf("Cisla boli uspesne nacistane zo suboru vstup.txt.\n");
    for (I=0;I<N;I++) {
        if ((I%2)==0) a[x]=pole[I];
        else a[x]=pole[I]+1;
        x++;
    }

    f=fopen("vstup.txt","w");
    fprintf(f,"A=[");
    for (I=0;I<x;I++) fprintf(f,"%d",a[I]);
    fprintf(f,"]");
    fclose(f);

    printf("\nVektor a bol uspesne ulozeny do suboru vstup.txt.\n\n");

    getch();
    return 0;
}
```

Príklad č. 29: Navrhните algoritmus, ktorý vyzve užívateľa k zadaniu prvkov matice A. Maticu uloží do súboru a vytvorí z nej maticu B tak, že k hodnote každého prvku pričíta index riadku, v ktorom sa daný prvok nachádza, teda hodnotu prvkov v 1. riadku zvýši o 1, v 2. riadku o 2 a pod. Maticu B program uloží do súboru matica.txt, pričom v súbore zachová aj prvky matice A.

Vstupné premenné: f - smerník na súbor (matica.txt)

M - počet riadkov, N - počet stĺpcov

Pomocná premenná: A[M][N]

Systémové premenné: I, J

Výstupná premenná: B[M][N]

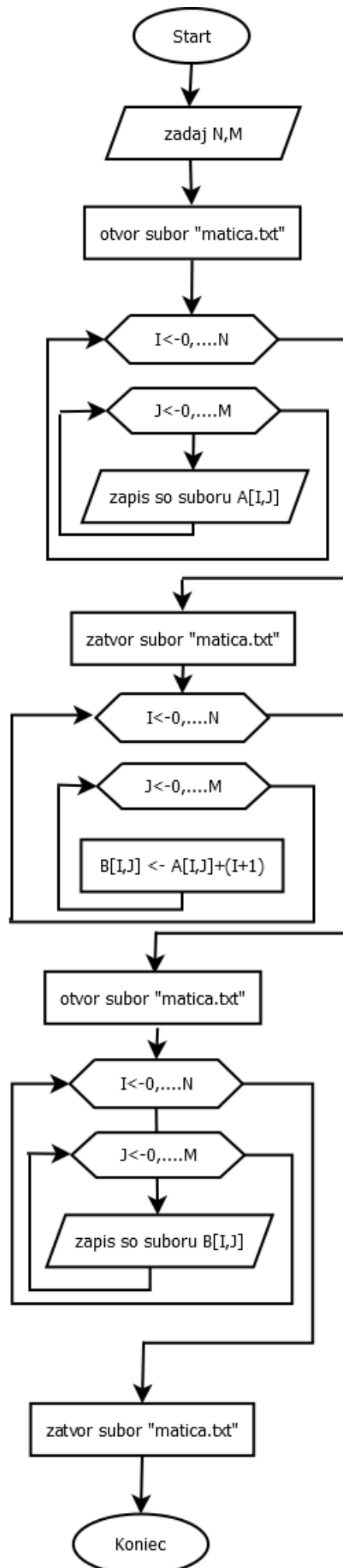
Metóda:

Užívateľ na vstupe zadá počet riadkov a stĺpcov. Potom zadá jednotlivé prvky matice. Matica je uložená do súboru matica.txt ako matica A. Ku každému prvku matice potom pričítame hodnotu $i+1$, čo zabezpečí, že napr. prvkom v 1. riadku bude pripočítaná hodnota 1 a nie 0. Výsledné prvky uložíme do matice B, ktorú uložíme do súboru matica.txt. Pri zapisovaní matice B do súboru použijeme mód „a“ namiesto „w“, čo zaručí zachovanie aj pôvodnej matice A.

Slovný popis algoritmu:

1. krok: zadaj počet riadkov a stĺpcov
2. krok: otvor súbor matica.txt
3. krok: ak i má hodnotu od 0 po M, pokračuj krokom 4, inak prejdí na krok 6
4. krok: ak j má hodnotu od 0 po N, pokračuj krokom 5, inak sa vráť na krok 3
5. krok: zapíš A[i][j] do súboru, vráť sa na krok 4
6. krok: zatvor súbor matica.txt
7. krok: ak i má hodnotu od 0 po M, prejdí na krok 8, inak prejdí na krok 11
8. krok: ak j má hodnotu od 0 po N, prejdí na krok 9, inak sa vráť na krok 7
9. krok: $B[i][j] \leftarrow A[i][j] + i + 1$
10. krok: otvor súbor matica.txt
11. krok: ak i má hodnotu od 0 po M, pokračuj krokom 12, inak prejdí na krok 14
12. krok: ak j má hodnotu od 0 po N, pokračuj krokom 13, inak prejdí na krok 11
13. krok: zapíš B[i][j] do súboru
14. krok: zatvor súbor matica.txt
15. krok: koniec

Vývojový diagram:



Prepis algoritmu v jazyku C:

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

int main() {

    FILE*f;
    int I,J,N,M;

    printf("Zadaj sirku matic:");
    scanf("%d",&M);
    printf("Zadaj vysku matic:");
    scanf("%d",&N);

    int a[M][N], b[M][N];

    f=fopen("matica.txt","w");
    fprintf(f,"Prvky matice a:\n");

    printf("Zadaj cisla matice a:\n");
    for (I=0;I<M;I++) {
        for (J=0;J<N;J++) {
            printf("a (%d,%d)=",I+1,J+1);
            scanf("%d",&a[I][J]);
            fprintf(f,"%d",a[I][J]);}
        fprintf(f,"\n");
    }
    fclose(f);
    printf("\nMatica a bola uspesne ulozena do suboru matica.txt.\n");

    for (I=0;I<M;I++) {
        for (J=0;J<N;J++) b[I][J]=a[I][J]+(I+1);
    }

    f=fopen("matica.txt","a");
    fprintf(f,"\n");
    fprintf(f,"Prvky matice b:\n");
    for (I=0;I<M;I++) {
        for (J=0;J<N;J++) fprintf(f,"%d",b[I][J]);
        fprintf(f,"\n");
    }

    fclose(f);

    printf("\nMatica b bola uspesne ulozena do suboru matica.txt.");

    getch();
    return 0;
}
```