**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 8.2 |
| **Autor** | Mateusz Jasiński | **Grupa** | WCY20IY2S1 |
| **Temat** | Pliki i strumienie – pliki tekstowe. | | |

1. Treść

Program wczytuje z klawiatury dwie nazwy pliku, a następnie odczytuje z pierwszego pliku liczby i wyświetla je na ekranie w wierszach po 10 liczb(funkcja), następnie sortuje te liczby(funkcja), ponownie wyświetla je na ekranie i zapisuje do drugiego pliku.

* 1. Metoda realizacji

Po wczytaniu nazw plików, program odczytuje elementy z pliku pierwszego, następnie zapisuje je w tablicy, sortuje bąbelkowo tablicę i posortowane elementy wypisuje na ekran oraz zapisuje do drugiego pliku.

* 1. Założenia / ograniczenia dotyczące danych:
     1. Dane wejściowe

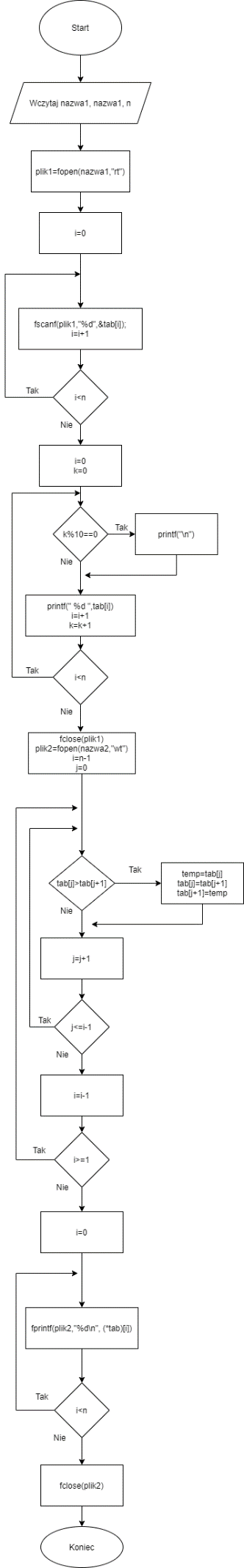
Nazwy plików – wprowadzane z klawiatury.

* + 1. Dane wyjściowe

n elementów przed i po sortowaniu - wyprowadzane na ekran

n elementów posortowanych - zapisywane do drugiego pliku.

1. Realizacja
   1. Algorytm



* 1. Kod źródłowy

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

bubble\_sort(int \*tab, int n) {

int i, j, temp;

for(i=n-1;i>=1;i--)

for(j=0;j<=i-1;j++)

if(tab[j]>tab[j+1]) {

temp=tab[j];

tab[j]=tab[j+1];

tab[j+1]=temp;

}

}

int \*wczytaj(char \*nazwa\_pliku1, int \*tab, int \*n) {

FILE\* plik=fopen(nazwa\_pliku1,"r");

if(!plik) {

printf("Nie znaleziono pliku do wczytania!");

exit(0);

}

int i;

while(fscanf(plik,"%\*d")!=EOF)

\*n=\*n+1;

rewind(plik);

tab=(int\*)malloc(\*n\*sizeof(int));

for(i=0;i<\*n;i++) {

fscanf(plik,"%d",&tab[i]);

printf("%d\t",tab[i]);

if((i+1)%10==0)

printf("\n");

}

fclose(plik);

return tab;

}

zapisz(char \*nazwa\_pliku, int \*tab, int n) {

FILE\* plik=fopen(nazwa\_pliku,"w");

if(!plik) {

printf("Nie udalo sie utworzyc pliku!");

exit(0);

}

int i;

for(i=0;i<n;i++)

fprintf(plik,"%d\n",tab[i]);

fclose(plik);

}

main() {

int \*tab,i,n=0;

char \*nazwa\_pliku1=(char\*)calloc(50, sizeof(char\*));

char \*nazwa\_pliku2=(char\*)calloc(50, sizeof(char\*));

printf("Podaj nazwe pliku do wczytania wartosci: ");

gets(nazwa\_pliku1);

printf("Podaj nazwe pliku do zapisania wartosci: ");

gets(nazwa\_pliku2);

tab=wczytaj(nazwa\_pliku1, tab, &n);

bubble\_sort(tab,n);

printf("\nPosortowane liczby:\n");

for(i=0;i<n;i++) {

printf("%d\t",tab[i]);

if((i+1)%10==0)

printf("\n");

}

zapisz(nazwa\_pliku2, tab, n);

free(nazwa\_pliku1);

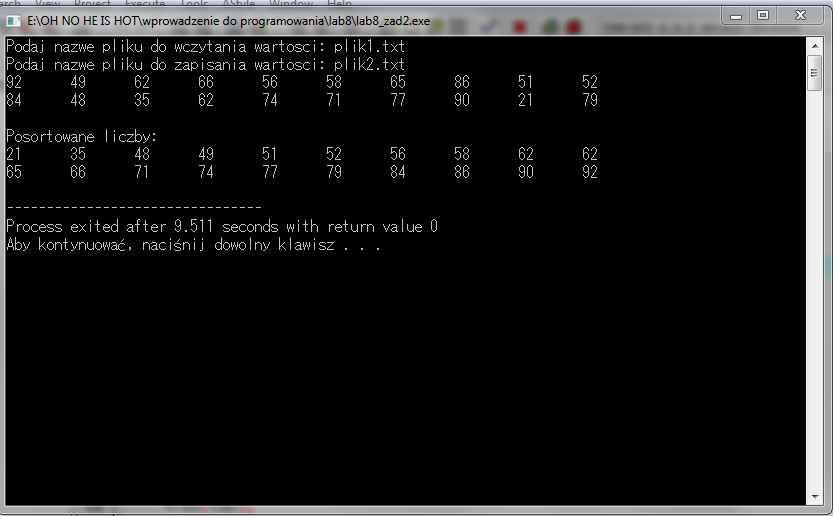
free(nazwa\_pliku2);

free(tab);

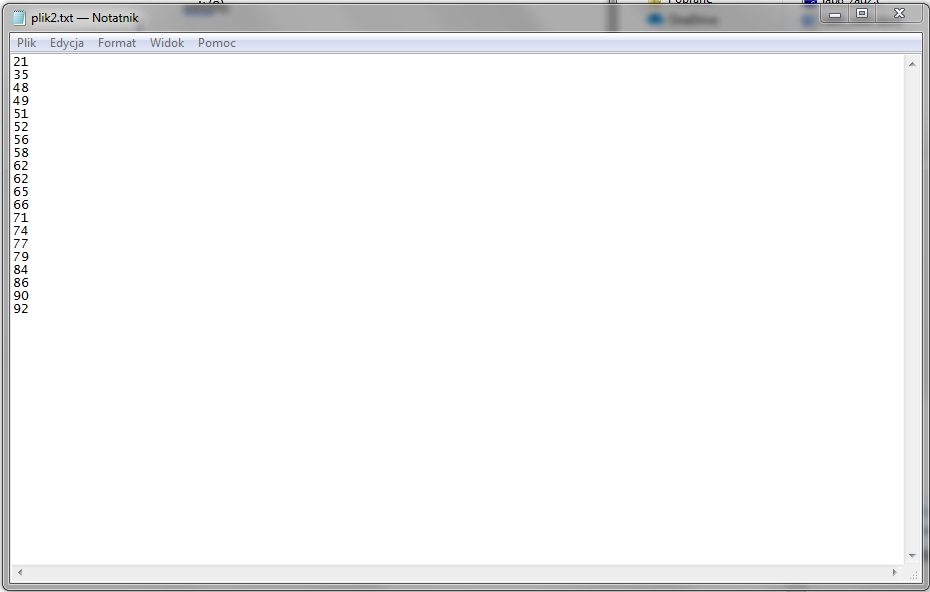
return 0;

}

* 1. Dane wejściowe



* 1. Dane wyjściowe



1. Wnioski

Złożoność obliczeniowa algorytmu:

O(n^2) = 3 + n + 1 + n + 1 + n^2 + 1 + n + 1 = n^2 + 3n + 7