**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 9.2 |
| **Autor** | Mateusz Jasiński | **Grupa** | WCY20IY2S1 |
| **Temat** | Pliki i strumienie – pliki binarne. | | |

1. Treść

Program wczytuje z pliku binarnego (plik z punktu 1) do tablicy rekordy, a następnie sortuje je wg pola int i zapisuje do kolejnego pliku listując na ekranie(dane każdego rekordu w kolejnym wierszu). Listowanie zaczyna nagłówek.

* 1. Metoda realizacji

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

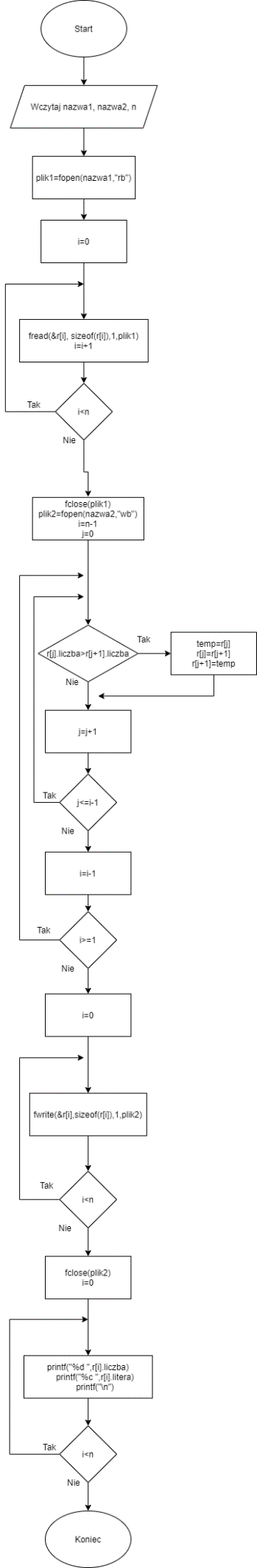
* 1. Założenia / ograniczenia dotyczące danych:
     1. Dane wejściowe

Nazwy plików – wprowadzane z klawiatury.

* + 1. Dane wyjściowe

n elementów posortowanych - wyprowadzane na ekran i zapisywane do drugiego pliku

1. Realizacja
   1. Algorytm



* 1. Kod źródłowy

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct pola {

int i;

char ch;

} pola;

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

int i, j;

pola \*temp;

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

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

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

\*temp=tab[j];

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

tab[j+1]=\*temp;

}

}

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

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

pola \*temp;

if(!plik) {

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

exit(0);

}

int i;

while(fread(&temp, sizeof(tab), 1, plik)!='\0')

\*n=\*n+1;

rewind(plik);

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

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

fread(tab+i, sizeof(tab), 1, plik);

fclose(plik);

return tab;

}

zapisz(char \*nazwa\_pliku, pola \*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++)

fwrite(tab+i, sizeof(tab), 1, plik);

fclose(plik);

}

main() {

int i,n=0;

pola \*tab;

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:\nint\tchar\n");

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

printf("%d\t%c\n",tab[i].i,tab[i].ch);

zapisz(nazwa\_pliku2, tab, n);

free(nazwa\_pliku1);

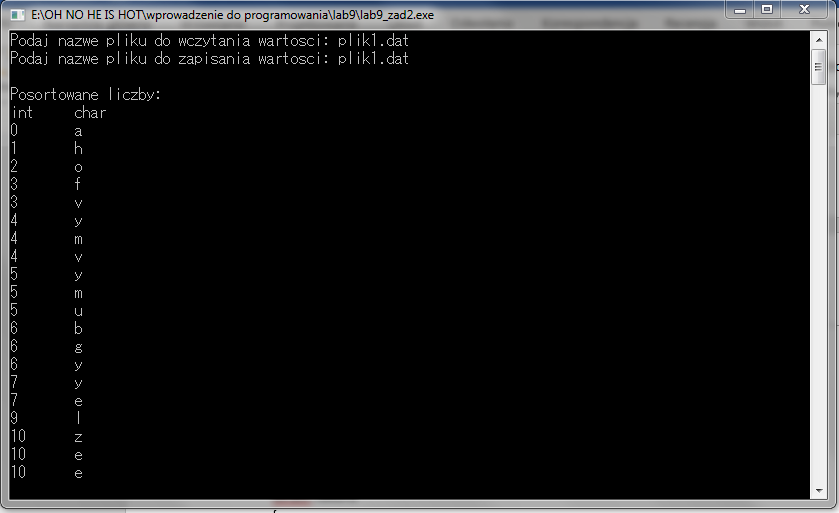
free(nazwa\_pliku2);

free(tab);

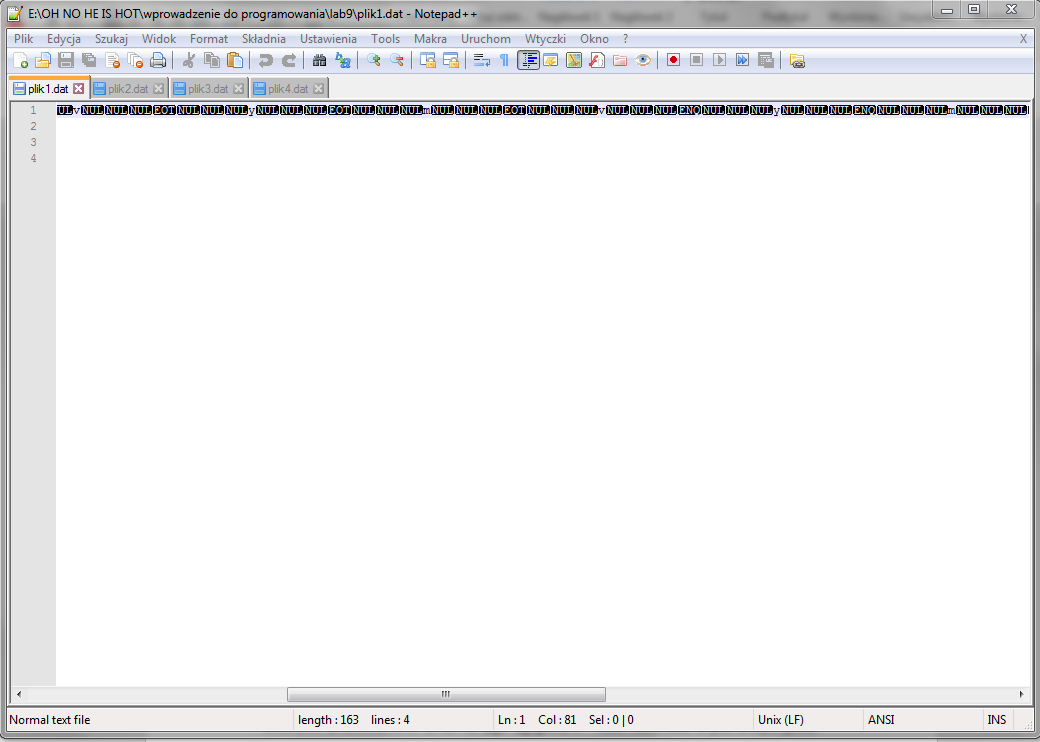
return 0;

}

* 1. Dane wejściowe



* 1. Dane wyjściowe



1. Wnioski

Złożoność obliczeniowa algorytmu:

O(n^2)=3+n+1+n^2+1+n+1+n=n^2+3n+6