

Zestaw 3A

Ocena:

ocena 3.0	Ocena 4.0	Ocena 5.0
Zad. 1 lub Zad. 2a	Zad. 1 i Zad. 2a lub Zadanie 2b	Zad. 1 i Zad. 2b

Ramówka programu na następnej stronie. Proszę z niej skorzystać. W razie potrzeby proszę odpowiednio zmodyfikować listy argumentów funkcji i typy wartości zwracanych przez funkcje.

Zadanie 1:

Zaimplementuj wybraną operację przywracania własności kopca max. Należy także dołączyć funkcję wyświetlającą stan kopca.

Zadanie 2:

Zaimplementuj algorytm BFS lub DFS dla reprezentacji grafu w postaci:

- macierzy sąsiedztwa
- listy sąsiedztwa (listę należy utworzyć na podstawie podanej w ramówce macierzy)

Wynikiem powinny być węzły w kolejności odwiedzenia.

Ramówka programu 3A:

```
#include <cstdlib>
#include <iostream>

using namespace std;

//Zadanie 1

const int K = 11; int kopiec[K] = {29,20,18,7,8,10,15,6,4,5,5}; //poprawny kopiec, należy zmiąć wybraną wartość
void przywracanie(int i){ //metoda .....
void print(){ //wydruk stanu kopca

void zadanie1(){
int i = ; //wybrany węzeł naruszający warunek kopca
cout<<"Zadanie 1"<<endl;      print(); przywracanie(i); print();      cout<<"Zadanie 1 - koniec"<<endl;
}

//Zadanie 2

const int N = 8;
int graf[N][N]={0,1,0,0,0,0,0,0}, {0,0,1,0,1,0,0,0}, {0,0,0,1,0,0,0,0}, {0,0,0,0,0,1,1,0}, {1,0,0,0,0,1,0,0},
{0,0,0,1,1,0,0,0}, {0,0,0,0,0,1,0,1}, {0,0,0,1,0,0,0,0}};
void przeszukiwanie(){ //algorytm .....
//listaS(){ //opcjonalna funkcja do utworzenia listy sąsiedztwa dla Zadania 2b

void zadanie2(){
cout<<"Zadanie 2"<<endl;      /*listaS();*/      przeszukiwanie();      cout<<"Zadanie 2 - koniec"<<endl;
}

int main(int argc, char *argv[])
{
zadanie1();      zadanie2();
system("PAUSE");
return EXIT_SUCCESS;
}
```