

## Zestaw 2A

Proszę skorzystać z podanej poniżej ramówki. W razie potrzeby proszę odpowiednio zmodyfikować listy argumentów funkcji i typy wartości zwracanych przez funkcje. Do każdego zadania należy przygotować funkcje umożliwiające wyświetlanie wyników (zgodnie z ramówką).

### Zadanie 1:

4pkt

Zaimplementuj operacje enqueue i dequeue dla kolejki FIFO reprezentowanej w postaci listy jednokierunkowej. Kolejka przechowuje znaki. Implementacja optymalna.

### Zadanie 2:

6pkt

Dla uporządkowanej niemalejąco listy dwukierunkowej zaimplementuj:

- operację utworzenia w/w listy, oraz
- operację usuwania elementu (jednego) o podanej wartości.

Lista przechowuje znaki. Implementacja optymalna.

### Ramówka do zadań:

```
//Zadanie 1
void enqueue(char z){}
char dequeue (){}
void print(){} //wydruk stanu kolejki
void zadanie1(){
cout<<"Zadanie 1"<<endl;
enqueue('a'); print(); enqueue('e'); print();enqueue('i'); print();enqueue('n'); print();enqueue('e'); print();
cout<< dequeue(); print(); cout<< dequeue(); print();
enqueue('r'); print();enqueue('x'); print();
cout<< dequeue(); print(); cout<< dequeue(); print(); cout<< dequeue(); print();
cout<< dequeue(); print(); cout<< dequeue(); print(); cout<< dequeue(); print(); cout<< dequeue(); print();
cout<<"Zadanie 1 - koniec"<<endl;
}

//Zadanie 2
void utworz(){} //dowolny sposób utworzenia listy, lista powinna zawierać następujące wartości: a,c,f,f, g,h,k, l
void usun(char z){}
void printLista(){} //wydruk listy
void zadanie2(){
cout<<"Zadanie 2"<<endl;
utworz(); printLista();
usun('f'); printLista();usun('h'); printLista();usun('l'); printLista();usun('a'); printLista();
cout<<"Zadanie 2 - koniec"<<endl;
}

int main(int argc, char *argv[])
{
    zadanie1();
    zadanie2();

    system("PAUSE");
    return EXIT_SUCCESS;
}
```