

### **Problema 3**

Creați template-urile și funcția necesară astfel încât execuția următorului cod:

```
void printDouble(const double& v) {
    std::cout << v << ",";
}
int main()
{
    DoubleLinkedList<double> d;
    d.PushOnFront(1.5);d.PrintAll(printDouble);
    d.PushOnFront(2.5);d.PushOnBack(3.5);d.PrintAll(printDouble);
    while (d.GetCount()) {
        std::cout << d.PopFromBack() << ",";
    }
    std::cout << std::endl;
    DoubleLinkedList<int> i;
    for (int index = 0; index < 10; index++) {
        if (index % 2 == 0)
            i.PushOnBack(index);
        else
            i.PushOnFront(index);
    }
    i.PrintAll(printInt);
    return 0;
}
```

să afișeze pe ecran următoarele:

```
Elements: 1 => 1.5,
Elements: 3 => 2.5,1.5,3.5,
3.5,1.5,2.5,
Elements: 10 => 9,7,5,3,1,0,2,4,6,8,
```

OBS/Cerinte:

- Nu aveți voie să folosiți template-uri din STL (de exemplu: vector, deque, etc) pentru rezolvarea acestei probleme.
- DoubleLinkedList ⇒ este o listă dublu înălțuită care folosește intern un template "Node" cu 3 câmpuri (next, precedent și value). Template-ul "Node" va fi definit într-un fișier header ("Node.h") separat.
- La adăugarea unui element în listă (prin metodele PushOnBack sau PushOnFront) se alocă un element de tipul Node care apoi este adăugat la listă prin câmpurile "next" și "precedent"
- Adăugați în fișierul main.cpp o funcție "printInt" care să afișeze un int (similar cu funcția printDouble). Funcția printInt este utilizată ca și parametru pentru metoda "PrintAll" apelată pentru variabila "i" din funcția main.

PS: Citiți pagina următoare pentru baremul de corectare

**Barem corectare (30 pct)**

Cerinta	Punctaj
Organizarea programului în 3 fișiere ("main.cpp" , "DoubleLinkedList", si "Node.h")	3 pct
Metoda <i>PushOnFront</i> din clasa parametrizată (template) DoubleLinkedList (se scade 1 pct dacă prototipul funcției nu folosește o referință constantă ci valoare pentru trimiterea parametrului).	5 pct
Metoda <i>PushOnBack</i> din clasa parametrizată (template) DoubleLinkedList (se scade 1 pct dacă prototipul funcției nu folosește o referință constantă ci valoare pentru trimiterea parametrului).	5 pct
Metoda <i>PopFromBack</i> din clasa parametrizată (template) DoubleLinkedList	5 pct
Metoda <i>PrintAll</i> din clasa parametrizată (template) DoubleLinkedList	3 pct
Constructorul din clasa parametrizată (template) DoubleLinkedList	1 pct
Destructorul din clasa parametrizată (template) DoubleLinkedList (care dealoca toate elementele din listă)	3 pct
Metoda <i>GetCount</i> din clasa parametrizată (template) DoubleLinkedList	1 pct
Funcția globală " <i>println</i> " adăugată în "main.cpp"	1 pct
Codul compilează și rulează corect	3 pct