

Zadanie 5

C-struktury (rozdziały 12 i 13).

Proszę napisać (i przetestować) opisany niżej program:

Struktura opisująca węzeł listy ma postać

```
struct Node {  
    int    data;  
    Node* next;  
};
```

(każdy węzeł przechowuje dane — w tym przypadku po prostu liczbę całkowitą `data`).
Napisać i przetestować następujące funkcje:

1. Funkcję o nagłówku

```
Node* arrayToSortedList(const int arr[], int size);
```

pobierającą tablicę `int`'ów i jej wymiar. Zadaniem funkcji jest utworzenie (operator `new`) listy jednokierunkowej obiektów struktury `Node`, zawierającej w kolejnych węzłach liczby z przekazanej (nieuporządkowanej) tablicy, ale w kolejności *wzrastającej*. Funkcja ma zwracać wskaźnik do „głowy” (pierwszego węzła) utworzonej listy.

2. Funkcję o nagłówku

```
void addSorted(Node*& head, int data);
```

pobierającą wskaźnik do „głowy” listy i dane (w postaci jednego `int`'a `data`). Zadaniem funkcji jest dodanie do listy węzła zawierającego dane `data` w taki sposób, aby w dalszym ciągu lista była uporządkowana. Wskaźnik do „głowy” (`head`) jest przekazany przez referencję, gdyż funkcja ta powinna móc go zmienić (gdy dotychczasowa lista była pusta, lub dana `data` ma wartość mniejszą niż ta zawarta w dotychczasowej „głowie”).

3. Funkcję o nagłówku

```
void deleteList(Node*& head);
```

usuwającą wszystkie węzły. Przy każdym usuwaniu węzła funkcja powinna drukować wartość danej `data` z usuwanego węzła, abyśmy widzieli, że rzeczywiście węzły te są usuwane. Po powrocie z funkcji, wartością `head` powinno być zero (`NULL`).

4. Funkcję o nagłówku

```
void printList(const Node* head);
```

wypisującą (w jednym wierszu) dane (`data`) z kolejnych węzłów listy (bez jej modyfikacji).

Przykładowy schemat programu:

```
#include <iostream>  
using namespace std;
```

```

struct Node {
    int    data;
    Node* next;
};

void printList(const Node* head) {
    // ...
}

void addSorted(Node*& head, int data) {
    // ...
}

Node* arrayToSortedList(const int arr[], int size) {
    // ...
}

void deleteList(Node*& head) {
    // ...
}

int main(void) {
    int arr[] = {7,4,8,3};
    int size = sizeof(arr)/sizeof(arr[0]);

    Node *head = arrayToSortedList(arr, size);
    printList(head);

    addSorted(head,6);
    addSorted(head,1);
    addSorted(head,9);
    printList(head);

    deleteList(head);
}

```

Program napisany według tego schematu powinien wydrukować coś w rodzaju:

```

3 4 7 8
1 3 4 6 7 8 9
Del 1; Del 3; Del 4; Del 6; Del 7; Del 8; Del 9;

```

Termin: do 4 maja (włącznie)

Rozwiązania, w postaci **jednego** pliku źródłowego zawierającego treść programu, proszę wrzucać w systemie EDU do katalogu „Foldery zadań / Zadanie_05”.

Nazwą pliku powinno być nazwisko z dużej litery (bez polskich znaków); rozszerzeniem musi być `'.cpp'`, czyli np. `Malinowska.cpp`.