

Laboratorium 5

Sprawdzian

- (2pkt) Dopisz brakujące fragmenty kodu jeśli zadany jest wynik:

```
int arr[] = {7,7,9,0};  
for ( .... : ... )  
    ... = ...  
for ( .... : ... ) std::cout << x << " ";  
// wynik to: 0 0 0 0
```

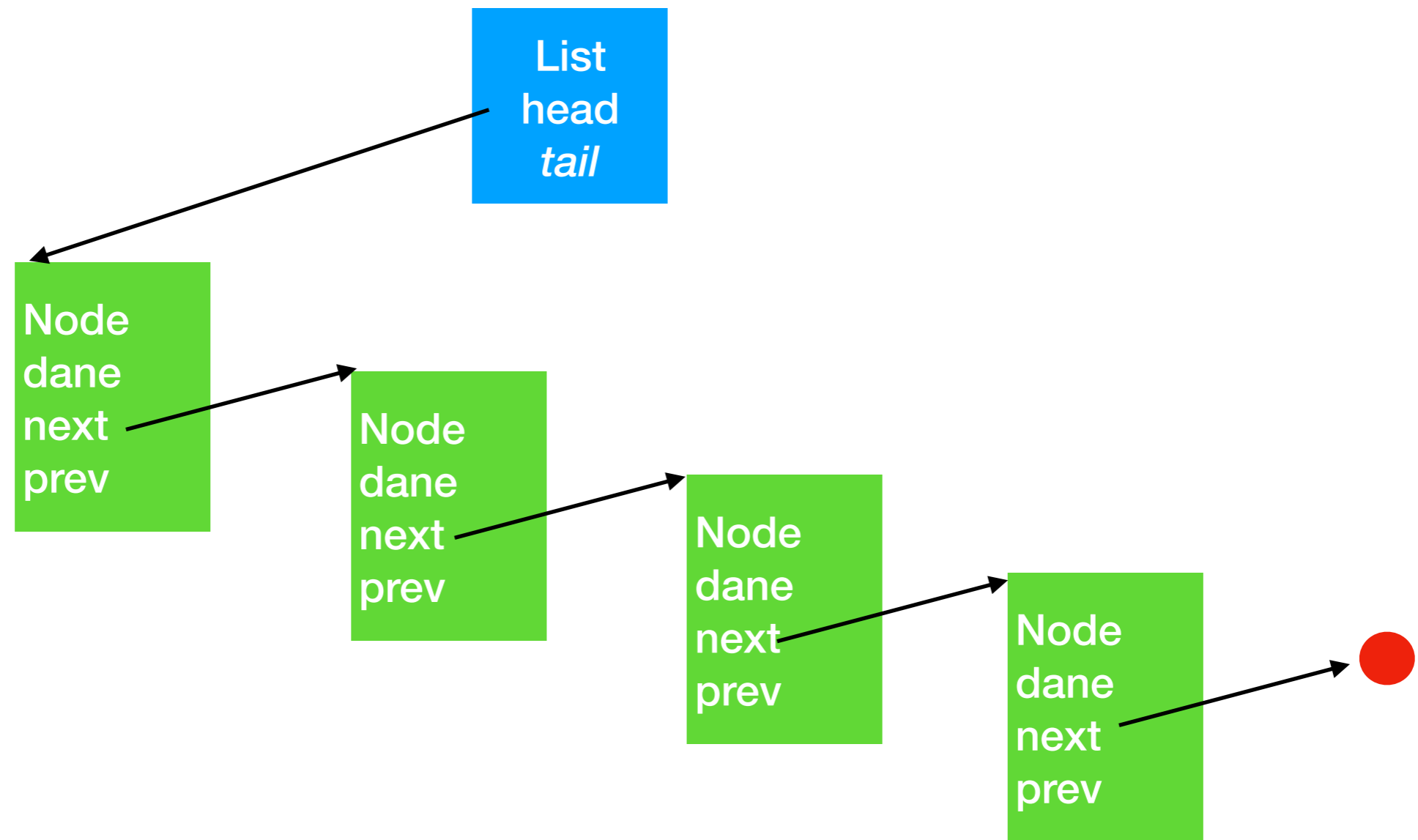
- (2pkt) Czy poniższy kod jest poprawny? Jeśli tak to jaki jest wynik działania poniższego programu jeśli z klawiatury zostanie wpisane 22:

```
void wczytaj1( int& x ) {  
    int & input = x;  
    std::cout << "Podaj liczbę:";  
    std::cin >> input;  
    return x;  
}  
int x = 7;  
int y = 4;  
int a = wczytaj(y);  
std::cout << x << " " << y << " " << a << "\n";
```

Zadanie

- Celem zadania jest napisanie **jednokierunkowej** listy połączonej służącej do przechowywania liczb całkowitych. Jest to wariant zadania, które było realizowane w poprzednim tygodniu z tym że **funkcje zamiast wskaźników muszą używać referencji** gdzie tylko to jest możliwe.
- (1pkt) `initList(lista);` ustawiająca strukturę List w stan “wyzerowany”
- (2pkt) `initList(lista, innaLista);` f. kopiująca z `innaLista`
- (2pkt) `endInsert(lista, x);` `frontInsert(lista, x);`
funkcje te odpowiednio dodają liczbę na początek i koniec listy
- (1pkt) `printList(list)` to funkcja, która wypisuje zawartość listy na ekran
- (1pkt) `resetList(list)` kasuje zawartość listy
- (2pkt) `addAfter(list, w, nowaLiczba)` to funkcja, która dodaje el. do listy po elemencie przechwującym wartość “w”, jeśli brak jest wartości w to dodaje na koniec listy
- (1pkt) `removeEl(list, w)` f. usuwa pierwszy element z listy, który jest równy w, jeśli go brak, nic nie robi
- Uwaga: w programie należy stworzyć dwie listy i dodać do nich po parę elementów i przetestować wszystkie funkcje

Lista wypełniona 4 elementami



head, tail, next, prev to wskaźniki na typ Node
dane to zmienna przechowująca nasze dane (int)