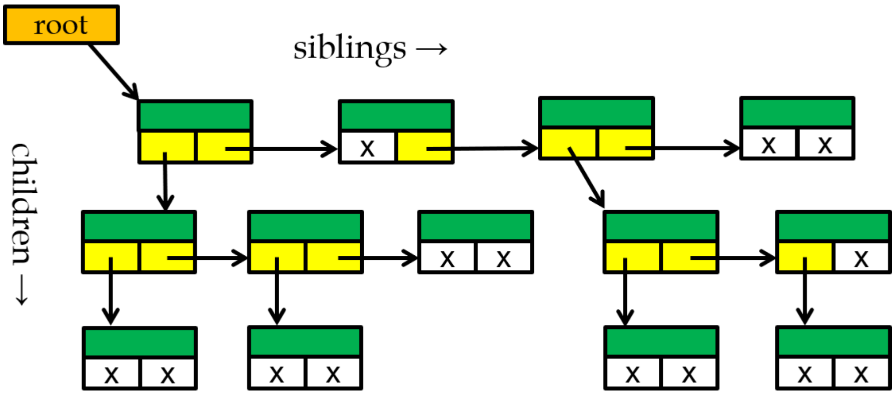
Exercise #5	Algorithms and Data Structures	
	Topic: Dynamic data structures – multiway trees	Version: 1.0 / 2019
	Prepared by: dr inż. Grzegorz Łukawski & dr inż. Barbara Łukawska	

1) Multiway trees

1.1) The definition

Multiway tree – a binary tree, where every node may have any number of children (children organized as a list). An example of a multiway tree:



1.2) Example implementation

Data structure:

```
struct Node {
        int id;
        string name;
        struct Node *child;
        struct Node *sibling;
};
```

Pointer to the first element of the tree (root):

```
struct Node *root = NULL;
```

Displaying a complete tree:

```
void display_tree(struct Node *t) {
        if (t != NULL) {
                cout << t->id << ": " << t->name << endl;
                display_tree(t->child);
                display_tree(t->sibling);
        }
}
```

Creating a new node:

```
struct Node *create_node(int id, string name) {
    struct Node *n = new Node;
    n->id = id;
    n->name = name;
    n->child = NULL;
    n->sibling = NULL;
    return(n);
}
```

Adding a new sibling to a node:

```
void add_sibling(struct Node *t, int id, string name) {
    if (t == NULL)    return;
    // Find the last sibling:
    while (t->sibling)      t = t->sibling;
    // Insert at the end of the list:
    t->sibling = create_node(id, name);
}
```

Adding a new child to a node:

```
void add_child(struct Node *t, int id, string name) {
    if (t == NULL)    return;
    // First or another child?
    if (t->child == NULL)   t->child = create_node(id, name);
    else                    add_sibling(t->child, id, name);
}
```

2) Exercises

Using the sample code shown above, write a program in **C++**, where the user can add new items* into a multiway tree and display the whole tree. Expand your program with the following features:

A) Adding a new item as a child or sibling of a given node (using its **id**)*;

B) Display all direct children or siblings of a given node (using its **id**);

Additional exercises:

C) Compute the min/max and average number of children in a family.

* list of children/siblings may be unordered.