Corrigé Type ASD 03

L= { 13, 17, 5, 21, 14, 22, 9, 20, 8, 6 }

|  |  |
| --- | --- |
| 13  / \  5 17  \ / \  9 14 21  / / \  8 20 22  /  6 |  |

postorderorder (Postfixe) : 6 8 9 5 14 20 22 21 17 13

preorder (prefixe) : 13 5 9 8 6 17 14 21 20 22

inorder (infixe ) : 5 6 8 9 13 14 17 20 21 22

breadth-First (Largeur) : 13 5 17 9 14 21 8 20 22 6

Exercice 02

bool rechercheElement(Node\* arbre, int v) {

if (arbre == nullptr)

return false;

if (arbre->value == v)

return true;

if (v < arbre->value)

rechercheElement(arbre->left, v);

else

rechercheElement(arbre->right, v);

}

Exercice 03

void bubbleSort vector<int>& array) {

int n = array.size();

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n - i - 1; ++j) {

if (array[j] > array[j + 1]) {

swap(array[j], array[j + 1]);

}

}

}

}

Node\* arrayToList(const std::vector<int>& array) {

if (array.empty()) {

return nullptr;

}

Node\* head = new Node(array[0]);

Node\* current = head;

for (size\_t i = 1; i < array.size(); ++i) {

current->next = new Node(array[i]);

current = current->next;

}

return head;

}

std::vector<int> listToArray(Node\* head) {

std::vector<int> array;

Node\* current = head;

while (current != nullptr) {

array.push\_back(current->data);

current = current->next;

}

return array;

}