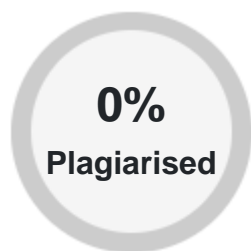


## PLAGIARISM SCAN REPORT



Excluded Url : None

### Content Checked For Plagiarism

```
NODE* trace_single_edge(SUFFIX_TREE* tree, NODE* node, PATH str, DBL_WORD* edge_pos, DBL_WORD* chars_found,
SKIP_TYPE type, int* search_done) { NODE* cont_node; DBL_WORD length, str_len; *search_done = 1; *edge_pos = 0; cont_node =
find_son(tree, node, tree->tree_string[str.begin]); if(cont_node == 0) { *edge_pos = get_node_label_length(tree, node)-1; *chars_found = 0;
return node; } node = cont_node; length = get_node_label_length(tree, node); str_len = str.end - str.begin + 1; if(type == skip) { if(length
tree_string[node->edge_label_start+*edge_pos] != tree->tree_string[str.begin+*edge_pos]) { (*edge_pos)--; return node; } } (*edge_pos)--;
if((*chars_found)
```