Contents

```
      1 Basic
      1

      1.1 ascii
      1

      1.2 limits
      1

      1.3 graph
      1

      2 Section2
      2

      2.1 thm
      2
```

1 Basic

1.1 ascii

```
1 int
             char
                        int
                                   char
                                              int
                                                        char
2
  32
                        64
                                              96
                                   @
3 | 33
                                              97
                        65
              !
                                   Α
                                                        а
 4 34
                        66
                                   В
                                              98
                                                        b
              #
5 35
                        67
                                   C
                                              99
6
   36
              $
                        68
                                   D
                                              100
                                                        d
7
   37
              %
                        69
                                   Ε
                                              101
                                   F
                                                        f
8 38
              &
                        70
                                              102
9 39
                        71
                                   G
                                              103
                                                        g
              (
10 40
                        72
                                   Н
                                              104
                                                        h
11
   41
                        73
                                   Ι
                                              105
                                                        i
12
   42
              *
                        74
                                   J
                                              106
13 43
                        75
                                   K
                                              107
                                                        k
14 44
                        76
                                   L
                                              108
                                   М
15 45
                        77
                                              109
                                                        m
16
   46
                        78
                                   Ν
                                              110
                                                        n
17 47
                        79
                                   0
                                              111
                                                        0
18 48
              0
                        80
                                              112
                                                        р
19 49
              1
                        81
                                   Q
                                              113
                                                        q
20 50
              2
                        82
                                   R
                                              114
21
  51
              3
                        83
                                   S
                                              115
                                                        s
22 52
              4
                        84
                                   Т
                                              116
                                                        t
23 53
              5
                        85
                                   U
                                              117
                                                        и
24 54
              6
                        86
                                   V
                                              118
25 55
              7
                        87
                                   W
                                              119
                                                        w
   56
              8
                        88
                                   Χ
                                              120
26
                                                        х
                                   γ
27 57
             9
                        89
                                              121
                                                        y
28 58
                        90
                                   Ζ
                                              122
                                                        z
29 59
                        91
                                   Γ
                                              123
                                                        {
30 60
                        92
                                              124
              <
                                   ١
31
  61
              =
                        93
                                   J
                                              125
                                                        }
32 62
                        94
                                              126
33 63
              ?
```

1.2 limits

```
[size]
1 [Type]
                                   [range]
2 char
                       1
                                 127 to -128
3 signed char
                       1
                                 127 to -128
4 unsigned char
                       1
                                 0 to 255
                                 32767 to -32768
5 short
                       2
6 int
                                 2147483647 to -2147483648
7
  unsigned int
                        4
                                 0 to 4294967295
                                 2147483647 to -2147483648
8 long
  unsigned long
                        4
                                 0 to 18446744073709551615
10 long long
                        8
11
              9223372036854775807 to -9223372036854775808
                             1.79769e+308 to 2.22507e-308
                       8
12 double
13 long double
                       16
                             1.18973e+4932 to 3.3621e-4932
                                3.40282e+38 to 1.17549e-38
14 float
                        4
15 unsigned long long
                                 0 to 18446744073709551615
                       8
                       32
16 string
```

1.3 graph

```
#include <bits/stdc++.h>
  using namespace std;
5
  class Node {
6
  public:
7
       int val:
       vector < Node *> children;
9
       Node() {}
10
11
       Node(int _val) {
12
13
           val = _val;
       }
14
15
       Node(int _val, vector<Node*> _children) {
16
17
           val = _val;
18
           children = _children;
19
       }
20
  };
21
22
  struct ListNode {
23
       int val;
       ListNode *next;
24
25
       ListNode(): val(0), next(nullptr) {}
26
       ListNode(int x) : val(x), next(nullptr) {}
       ListNode(int x, ListNode *next) : val(x),
27
           next(next) {}
28 };
29
  struct TreeNode {
30
31
       int val;
       TreeNode *left;
32
33
       TreeNode *right;
34
       TreeNode() : val(0), left(nullptr),
           right(nullptr) {}
35
       TreeNode(int x) : val(x), left(nullptr),
           right(nullptr) {}
       TreeNode(int x, TreeNode *left, TreeNode *right)
36
           : val(x), left(left), right(right) {}
37 };
38
  class ListProblem {
39
       vector<int> nums={};
40
  public:
41
42
       void solve() {
43
           return;
44
       }
45
       ListNode* buildList(int idx) {
46
47
           if(idx == nums.size()) return NULL;
48
           ListNode *current=new
               ListNode(nums[idx++], current ->next);
49
           return current;
       }
50
52
       void deleteList(ListNode* root) {
           if(root == NULL) return;
53
54
           deleteList(root->next);
55
           delete root:
56
           return;
       }
57
58
  };
59
60
  class TreeProblem {
61
       int null = INT_MIN;
       vector<int> nums = {}, result;
  public:
64
       void solve() {
65
66
           return;
67
       }
68
       TreeNode* buildBinaryTreeUsingDFS(int left, int
69
           right) {
70
           if((left > right) || (nums[(left+right)/2] ==
                null)) return NULL;
```

```
71
            int mid = (left+right)/2;
72
            TreeNode* current = new TreeNode(
                                                                  143 };
73
                 nums[mid],
                                                                  144
                 buildBinaryTreeUsingDFS(left,mid-1),
                                                                  145 int main() {
74
                 buildBinaryTreeUsingDFS(mid+1, right));
75
                                                                  146
76
            return current;
                                                                  147
77
                                                                  148 }
78
79
        TreeNode* buildBinaryTreeUsingBFS() {
80
            int idx = 0;
81
            TreeNode* root = new TreeNode(nums[idx++]);
            queue < TreeNode *> q;
82
83
            q.push(root);
            while(idx < nums.size()) {</pre>
84
85
                 if(nums[idx] != null) {
                     TreeNode* left = new
86
                          TreeNode(nums[idx]);
87
                     q.front()->left = left;
                     q.push(left);
88
                 }
89
                 idx++:
90
91
                 if((idx < nums.size()) && (nums[idx] !=</pre>
                      null)) {
                     TreeNode* right = new
92
                          TreeNode(nums[idx]);
                     q.front()->right = right;
93
                     q.push(right);
94
                 }
95
                 idx++;
96
97
                 q.pop();
98
99
            return root;
100
       }
101
        Node* buildNAryTree() {
102
103
            int idx = 2;
104
            Node *root = new Node(nums.front());
            queue < Node *> q;
105
            q.push(root);
106
107
            while(idx < nums.size()) {</pre>
                 while((idx < nums.size()) && (nums[idx]</pre>
108
                      != null)) {
                     Node *current = new Node(nums[idx++]);
109
110
                     q.front()->children.push_back(current);
111
                     q.push(current);
112
                 }
113
                 idx++:
                 q.pop();
114
            }
115
116
            return root;
117
118
119
        void deleteBinaryTree(TreeNode* root) {
120
            if(root->left != NULL)
                 deleteBinaryTree(root->left);
            if(root->right != NULL)
121
                 deleteBinaryTree(root->right);
            delete root;
122
123
            return;
124
125
        void deleteNAryTree(Node* root) {
126
            if(root == NULL) return;
127
            for(int i=0; i<root->children.size(); i++) {
128
129
                 deleteNAryTree(root->children[i]);
130
                 delete root->children[i];
131
132
            delete root;
133
            return;
       }
134
135
        void inorderTraversal(TreeNode* root) {
136
137
            if(root == NULL) return;
            inorderTraversal(root->left);
138
            cout << root -> val << ' ';</pre>
139
140
            inorderTraversal(root->right);
141
            return;
```

Section2 2

return 0;

2.1 thm

}

- 中文測試
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$