









Data Organization Ex3

 Assignee	 Viet
 Status	Done
 Due	@March 23, 2024
 Project	 <u>HCMUS</u>
 Priority	High
 Spent time (Hours)	4

Thông tin sinh viên

- Họ và tên: Cao Hoài Việt
- MSSV: 22850034
- Email SV: 22850034@student.hcmus.edu.vn
- Email cá nhân: viet.ch2612@gmail.com
- Github: <https://github.com/vietch2612>

Bài làm

Câu 1. Module về thời gian trong ngày

Module `Time.h`

```
#include <iostream>
#include <sstream>

class Time {
private:
    int hour, minute, second;

public:
    // Validate the time format
    bool isValid() {
        bool isHourValid = false;
```

```

    bool isMinuteValid = false;
    bool isSecondValid = false;
    if (hour >= 0 && hour < 24) isHourValid = true;
    if (minute >= 0 && minute < 60) isMinuteValid = true;
    if (second >= 0 && second < 60) isSecondValid = true;

    return isHourValid && isMinuteValid && isSecondValid;
}

// C++ way to parse string instead of scanf
static Time parseFromString(std::string& timeString) {
    Time t;
    std::stringstream ss(timeString);
    char discard;
    ss >> t.hour >> discard >> t.minute >> discard >> t.second;
    return t;
}

// Convert all to seconds and return sum
int toSeconds() { return hour * 3600 + minute * 60 + second; }

// Convert to seconds then compare
static int secondDiff(Time& t1, Time& t2) {
    return t1.toSeconds() - t2.toSeconds();
}

// If this.time == anotherTime return 0;
// If this.time > anotherTime return -1;
// If this.time < anotherTime return 1;
int compareTo(Time& anotherTime) {
    if (toSeconds() == anotherTime.toSeconds()) return 0;
    return toSeconds() < anotherTime.toSeconds() ? -1 : 1;
}
};

```

time.cpp

```
#include "Time.h"
```

```

#include <iostream>

int main() {
    std::string timeString1, timeString2;

    std::cout << "Nhap thoi diem thu nhat HH:MM:SS: ";
    std::cin >> timeString1;
    Time time1 = Time::parseFromString(timeString1);
    std::cout << "Nhap thoi diem thu hai HH:MM:SS: ";
    std::cin >> timeString2;
    Time time2 = Time::parseFromString(timeString2);

    if (!time1.isValid() && !time2.isValid()) {
        std::cout << "Thoi gian khong hop le" << std::endl;
        return 1;
    }
    std::cout << "Thoi gian hop le." << std::endl;

    int secondDiff = Time::secondDiff(time1, time2);
    if (secondDiff == 0) {
        std::cout << "Hai thoi diem nay trung nhau";
    } else if (secondDiff > 0) {
        std::cout << "Thoi diem thu 2 dien ra truoc";
    } else {
        std::cout << "Thoi diem thu nhat dien ra truoc";
    }
    std::cout << std::endl;

    std::cout << "Tong so giay giua hai thoi diem: "
                << std::abs(secondDiff) << std::endl;

    return 0;
}

```

Kết quả thực thi

```
< x binary tree.cpp x zsh x linked_list x linked_list.cpp x bag.cpp x Time.h x
1 #include <iostream>
2 #include <sstream>
3
4 class Time {
5 private:
6     int hour, minute, second;
7
8 public:
9     // Validate the time format
10    bool isValid() {
11        bool isHourValid = false;
12        bool isMinuteValid = false;
13        bool isSecondValid = false;
14        if (hour >= 0 && hour < 24) isHourValid = true;
15        if (minute >= 0 && minute < 60) isMinuteValid = true;
16        if (second >= 0 && second < 60) isSecondValid = true;
17        return isHourValid && isMinuteValid && isSecondValid;
18    }
19
20    // Easy way to parse string instead of scanf
21    static Time parseFromString(std::string& timeString) {
22        Time t;
23        std::stringstream ss(timeString);
24        char discard;
25        ss >> t.hour >> discard >> t.minute >> discard >> t.second;
26        return t;
27    }
28
29    // Convert all to seconds and return sum
30    int toSeconds() { return hour * 3600 + minute * 60 + second; }
31
32    // Convert to seconds then compare
33    static int secondDiff(Time& t1, Time& t2) {
34        return t1.toSeconds() - t2.toSeconds();
35    }
36
37    // If this.time == anotherTime return 0;
38    // If this.time > anotherTime return -1;
39    // If this.time < anotherTime return 1;
40    int compareTo(Time& anotherTime) {
41        if (toSeconds() == anotherTime.toSeconds()) return 0;
42        return toSeconds() < anotherTime.toSeconds() ? -1 : 1;
43    }
44 };
45
88 ex3 >>> clang++ time.cpp -o time
89 ex3 >>> ./time
90 Nhập thời điểm thu nhất HH:MM:SS: 10:10:10
91 Nhập thời điểm thu hai HH:MM:SS: 20:20:30
92 Thời gian hợp lệ.
93 Thời điểm thu nhất diện ra trước
94 Tổng số giây giữa hai thời điểm: 36620
95 ex3 >>> ./time
96 Nhập thời điểm thu nhất HH:MM:SS: 24:00:00
97 Nhập thời điểm thu hai HH:MM:SS: 13:11:00
98 Thời gian hợp lệ.
99 Thời điểm thu 2 diện ra trước
100 Tổng số giây giữa hai thời điểm: 38940
101 ex3 >>> ./time
102 Nhập thời điểm thu nhất HH:MM:SS: 13:00:00
103 Nhập thời điểm thu hai HH:MM:SS: 24:00:00
104 Thời gian hợp lệ.
105 Thời điểm thu nhất diện ra trước
106 Tổng số giây giữa hai thời điểm: 39600
107 ex3 >>> ./time
108 Nhập thời điểm thu nhất HH:MM:SS: 10:00:00
109 Nhập thời điểm thu hai HH:MM:SS: 10:01:00
110 Thời gian hợp lệ.
111 Thời điểm thu nhất diện ra trước
112 Tổng số giây giữa hai thời điểm: 60
113 ex3 >>> ./time
114 Nhập thời điểm thu nhất HH:MM:SS: 10:00:00
115 Nhập thời điểm thu hai HH:MM:SS: 9:0:0
116 Thời gian hợp lệ.
117 Thời điểm thu 2 diện ra trước
118 Tổng số giây giữa hai thời điểm: 3600
119 ex3 >>>
120
121
122
123
124
125
126
127
128
TERMINAL zsh
~/Documents/hcm-assignment/data_organization/ex3/time.cpp 35L, 943B Written data_organization 92%
```

Câu 2. Mô hình túi dữ liệu (bag).

```
#include <iostream>

#define NOT_A_VALUE -1
#define MAX_VALUE 1000

class Bag {
private:
```

```

int size = 0;
int values[MAX_VALUE] = {NOT_A_VALUE};

void removeAt(int index) { values[index] = NOT_A_VALUE; }

void reOrder() {
    int j = 0;
    for (int i = 0; i < size; i++) {
        if (values[i] != NOT_A_VALUE) {
            values[j] = values[i];
            j++;
        }
    }
    size = j;
}

public:
    Bag() {}

    // Insert a new value
    void insert(int value) { values[size++] = value; };

    // Remove a single element
    void remove(int value) {
        for (int i = 0; i < size; i++) {
            if (value == values[i]) {
                removeAt(i);
                break;
            }
        };
        reOrder();
    }

    // Remove all
    void removeAll(int value) {
        for (int i = 0; i < size; i++)
            if (value == values[i]) removeAt(i);
        reOrder();
    }

```

```

};

// Count
int count(int value) {
    int c = 0;
    for (int i = 0; i < size; i++)
        if (values[i] == value) c++;
    return c;
}

// Check if two bags are equal
bool isEqual(Bag& aBag) {
    // Check the size if equal
    if (size != aBag.size) return false;

    // Init an array to store the count of each value
    int count[MAX_VALUE] = {0};
    // Then count the number of each value in this.bag
    for (int i = 0; i < size; i++) count[values[i]]++;
    // Then decrease the count
    // of each value in another bag aBag
    for (int i = 0; i < size; i++) count[aBag.values[i]]--;
    // If all the count is 0, then the two bags are equal
    for (int i = 0; i < size; i++)
        if (count[i] != 0) return false;
    return true;
}

// Check if this bag is subset of another bag
// This way, it would take more time than
// the way I'm using in isEqual()
bool isSubsetOf(Bag& aBag) {
    for (int i = 0; i < size; i++)
        if (count(values[i]) > aBag.count(values[i])) return false;
    return true;
}

// Merge two bags

```

```

void merge(Bag& aBag) {
    for (int i = 0; i < aBag.size; i++)
        values[size++] = aBag.values[i];
};
};
};

```

Kết quả thực thi

```

108 bag1.print();
109 std::cout << std::endl;
110
111 std::cout << "Xoa so 1: ";
112 bag1.remove(1);
113 bag1.print();
114 std::cout << std::endl;
115
116 std::cout << "Xoa toan bo so 5: ";
117 bag1.removeAll(5);
118 bag1.print();
119 std::cout << std::endl;
120
121 std::cout << "So lan xuat hien cua so 1: " << bag1.count(1) << std::endl;
122 std::cout << "So lan xuat hien cua so 5: " << bag1.count(5) << std::endl;
123 std::cout << "So lan xuat hien cua so 7: " << bag1.count(7) << std::endl;
124 std::cout << std::endl;
125
126 Bag bag2;
127 Bag bag3;
128 bag2 = bag1;
129 bag3 = bag1;
130 bag3.remove(7);
131 std::cout << "Tui 1: ";
132 bag1.print();
133 std::cout << "Tui 2: ";
134 bag2.print();
135 std::cout << "Tui 3: ";
136 bag3.print();
137 std::cout << "Tui 1 co bang tui 2 khong?: " << bag1.isEqual(bag2)
138 << std::endl;
139 std::cout << "Tui 1 co bang tui 3 khong?: " << bag1.isEqual(bag3)
140 << std::endl;
141 std::cout << "Tui 1 co co phai tui con cua 2 khong?: "
142 << bag1.isSubsetOf(bag2) << std::endl;
143 std::cout << "Tui 1 co co phai tui con cua 3 khong?: "
144 << bag1.isSubsetOf(bag3) << std::endl;
145 std::cout << "Tui 3 co co phai tui con cua 1 khong?: "
146 << bag3.isSubsetOf(bag1) << std::endl;
147 std::cout << std::endl;
148
149 Bag bag4;
150 bag4.insert(8);
151 bag4.insert(20);
152 std::cout << "Tui 4: ";
153 bag4.print();
154 std::cout << "Tui 1 + tui 4 se la: ";
155 bag1.merge(bag4);
156 bag1.print();
157
158 return 0;

```

```

64 ex3 >>> clang++ bag.cpp -o bag 66 ./bag
65 Tui 1: 1, 2, 1, 3, 3, 5, 5, 7, 7, 7,
66
67 Them mot so 4: 1, 2, 1, 3, 3, 5, 5, 7, 7, 7, 4,
68
69 Xoa so 1: 2, 1, 3, 3, 5, 5, 7, 7, 4,
70
71 Xoa toan bo so 5: 2, 1, 3, 3, 7, 7, 7, 4,
72
73 So lan xuat hien cua so 1: 1
74 So lan xuat hien cua so 5: 0
75 So lan xuat hien cua so 7: 3
76
77 Tui 1: 2, 1, 3, 3, 7, 7, 7, 4,
78 Tui 2: 2, 1, 3, 3, 7, 7, 7, 4,
79 Tui 3: 2, 1, 3, 3, 7, 7, 4,
80 Tui 1 co bang tui 2 khong?: 1
81 Tui 1 co bang tui 3 khong?: 0
82 Tui 1 co co phai tui con cua 2 khong?: 1
83 Tui 1 co co phai tui con cua 3 khong?: 0
84 Tui 3 co co phai tui con cua 1 khong?: 1
85
86 Tui 4: 8, 20,
87 Tui 1 + tui 4 se la: 2, 1, 3, 3, 7, 7, 7, 4, 8, 20,
88 ex3 >>>

```

64 ex3 >>> clang++ bag.cpp -o bag 66 ./bag
 65 Tui 1: 1, 2, 1, 3, 3, 5, 5, 7, 7, 7,
 66
 67 Them mot so 4: 1, 2, 1, 3, 3, 5, 5, 7, 7, 7, 4,
 68
 69 Xoa so 1: 2, 1, 3, 3, 5, 5, 7, 7, 4,
 70
 71 Xoa toan bo so 5: 2, 1, 3, 3, 7, 7, 7, 4,
 72
 73 So lan xuat hien cua so 1: 1
 74 So lan xuat hien cua so 5: 0
 75 So lan xuat hien cua so 7: 3
 76
 77 Tui 1: 2, 1, 3, 3, 7, 7, 7, 4,
 78 Tui 2: 2, 1, 3, 3, 7, 7, 7, 4,
 79 Tui 3: 2, 1, 3, 3, 7, 7, 4,
 80 Tui 1 co bang tui 2 khong?: 1
 81 Tui 1 co bang tui 3 khong?: 0
 82 Tui 1 co co phai tui con cua 2 khong?: 1
 83 Tui 1 co co phai tui con cua 3 khong?: 0
 84 Tui 3 co co phai tui con cua 1 khong?: 1
 85
 86 Tui 4: 8, 20,
 87 Tui 1 + tui 4 se la: 2, 1, 3, 3, 7, 7, 7, 4, 8, 20,
 88 ex3 >>>

Câu 3. Viết hàm `int sumPos(NODE *head)` để tính tổng các số dương trong danh sách liên kết có đầu trỏ bởi head bằng kĩ thuật lặp và đệ qui.

3.1 Sử dụng kĩ thuật lặp

```
// Traverse from the head toward the last and sum
int sumPosByIterative(NODE* head) {
    int total = 0;
    NODE* c = head;
    while (c != NULL) {
        total += c->data > 0 ? c->data : 0;
        c = c->next;
    }

    return total;
}
```

3.2 Sử dụng kĩ thuật đệ qui

```
// Recursive
int sumPosByRecursive(NODE* head) {
    if (head == NULL) return 0;
    return (head->data > 0 ? head->data : 0)
        + sumPosByRecursive(head->next);
}
```

Kết quả thực thi

```
[1] -> [3] -> [-1] -> [8] -> [-2] -> [4]
// Expected: 16
```



```

22 while (last->next != NULL) {
23     last = last->next;
24 }
25 last->next = new_node;
26 };
27
28 // Traverse from head toward the last and sum
29 int sumPosByIterative(NODE* head) {
30     int total = 0;
31     NODE* c = head;
32     while (c != NULL) {
33         total += c->data > 0 ? c->data : 0;
34         c = c->next;
35     }
36     return total;
37 }
38
39 // Recursive
40 int sumPosByRecursive(NODE* head) {
41     if (head == NULL) return 0;
42     return (head->data > 0 ? head->data : 0) + sumPosByRecursive(head->next);
43 }
44
45 int main() {
46     NODE* head = NULL;
47     append(&head, 1);
48     append(&head, 3);
49     append(&head, -1);
50     append(&head, 8);
51     append(&head, -2);
52     append(&head, 4);
53     std::cout << "Tong (Su dung lap): " << sumPosByIterative(head) << std::endl;
54     std::cout << "Tong (Su dung de quy): " << sumPosByRecursive(head)
55     << std::endl;
56     return 0;
57 }
58
1 ex3 >>> clang++ linked_list.cpp -o linked_list
2 ex3 >>> ./linked_list
3 Tong (Su dung lap): 16
4 Tong (Su dung de quy): 16
5 ex3 >>>

```

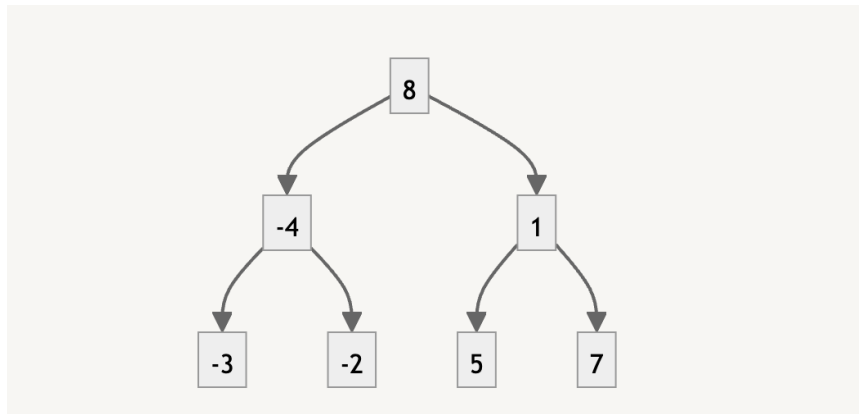
Câu 4. Viết hàm `int sumPos(NODE *root)` để tính các số dương của cây nhị phân.

```

int sumPos(NODE* root) {
    if (root == NULL) return 0;
    return (root->data > 0 ? root->data : 0) + sumPos(root->left)
        + sumPos(root->right);
};

```

Kết quả thực thi với cây sau

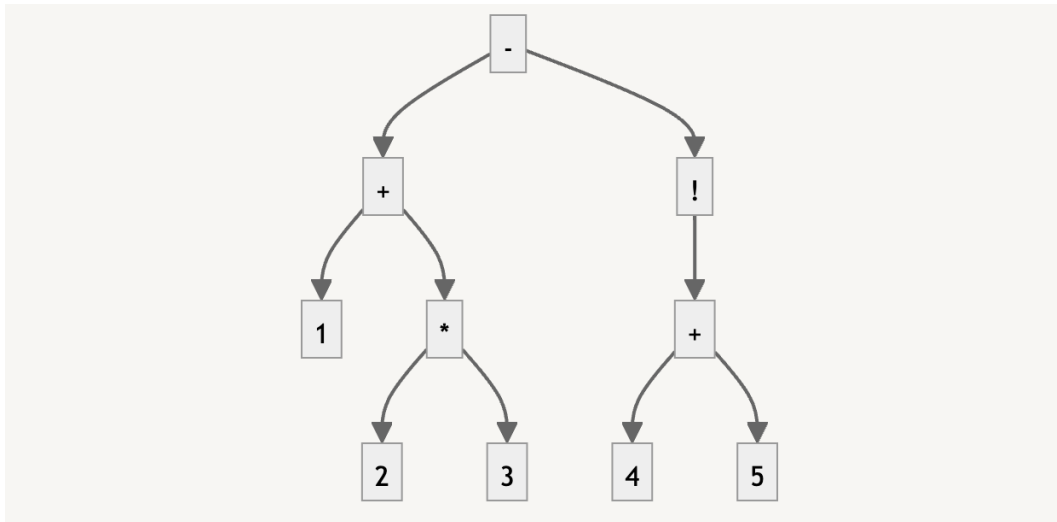


```
1 #include <iostream>
2
3 struct NODE {
4     int data;
5     NODE *left, *right;
6 };
7
8 // Sum using recursive
9 int sumPos(NODE* root) {
10     if (root == NULL) return 0;
11     return (root->data > 0 ? root->data : 0) + sumPos(root->left) +
12         sumPos(root->right);
13 };
14
15 // Support method
16 // To create a new node faster
17 NODE* newNode(int value) {
18     NODE* node = new NODE();
19     node->data = value;
20     return node;
21 }
22
23 int main() {
24     // Create a new node to test
25     NODE* root = newNode(8);
26     root->left = newNode(-4);
27     root->right = newNode(1);
28     root->left->left = newNode(-3);
29     root->left->right = newNode(-2);
30     root->right->left = newNode(5);
31     root->right->right = newNode(7);
32
33     std::cout << "Tong cua cay = " << sumPos(root) << std::endl;
34 }
35
```

```
1 ex3 >>> clang++ binary_tree.cpp -o binary_tree 66 ./binary_tree
2 Tong cua cay = 21
3 ex3 >>>
```

Câu 5. Cây biểu thức số học

A. Vẽ cây biểu thức số học của biểu thức $1 + 2 * 3 - (4 + 5)!$



B. Duyệt trước

- + 1 * 2 3 ! + 4 5

C. Duyệt sau

1 2 3 * + 4 5 + ! -