

BÁO CÁO THỰC HÀNH TUẦN 40
HỌC PHẦN: IT3040 - KỸ THUẬT LẬP TRÌNH - 20221

Mục lục

Bài thực hành số 4

Bài 4.1: Đảo ngược một danh sách liên kết đơn.....	1
4.2 Một điểm trong không gian 2 chiều được biểu diễn bằng pair. Hãy viết hàm tính diện tích tam giác theo tọa độ 3 đỉnh.....	4
4.3 Một vector trong không gian 3 chiều được biểu diễn bằng tuple<double, double, double>. Hãy viết hàm tính tích có hướng của 2 vector	5
Bài 4.4. Cho hai std::vector, hãy xóa hết các phần tử chẵn, sắp xếp giảm dần các số trong cả 2 vector và trộn lại thành một vector cũng được sắp xếp giảm dần.....	6
Bài 4.5. Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề. Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán DFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề. Yêu cầu hàm trả ra thứ tự các đỉnh được thăm (những đỉnh không thể thăm từ đỉnh 1 thì không phải in ra).	9
Bài 4.6. Viết hàm void bfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề. Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán BFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề. Yêu cầu hàm trả ra	10
Bài 4.7. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set.....	11
Bài 4.8. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.	13
Bài 4.9. Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh	15
Bài tập 10: Search Engi.....	17
Bài tập 11. Bảo vệ lâu đài.....	20
Bài tập 12. Lược đồ	23
Bài tập 13: Đếm xâu con	24

Bài 4.1: Đảo ngược một danh sách liên kết đơn

Time taken 4 mins 14 secs
Grade 100.00 out of 100.00

Question 1
Correct
Mark 15.00 out of 15.00
Flag question

Bài 4.1: Đảo ngược một danh sách liên kết đơn

Hãy hoàn thiện các hàm thao tác trên một danh sách liên kết:

- Thêm một phần tử vào đầu danh sách liên kết
- In danh sách
- Đảo ngược danh sách liên kết (yêu cầu độ phức tạp thời gian $O(N)$ và chi phí bộ nhớ dùng thêm $O(1)$)

For example:

Input	Result
10 -1 4 5 7 2 4 6 7 12 50	Original list: 50 12 7 6 4 2 7 5 4 -1 Reversed list: -1 4 5 7 2 4 6 7 12 50

Answer: (penalty regime: 10, 20, ... %)

```

1 #include <iostream>
2 using namespace std;
3 struct Node {
4     int data;

```

Answer: (penalty regime: 10, 20, ... %)

```

4     int data;
5     Node* next;
6
7     Node(int data) {
8         this->data = data;
9         next = NULL;
10    }
11 };
12
13 // push a new element to the beginning of the list
14 Node* prepend(Node* head, int data) {
15
16     Node* new_node = new Node(data);
17     new_node->next = head;
18
19     head = new_node;

```

Answer: (penalty regime: 10, 20, ... %)

```

24
25 // print the list content on a line
26 void print(Node* head) {
27
28     Node* tmp = head;
29     while(tmp != NULL){
30         printf("%d ", tmp->data);
31         tmp = tmp->next;
32     }
33     cout << endl;
34     /*****/
35 }
36
37 // return the new head of the reversed list
38 Node* reverse(Node* head) {
39     /*****/
40     Node* current = head->next;
41     Node* prev = head; prev->next = NULL;
42     Node* next = NULL;
43

```

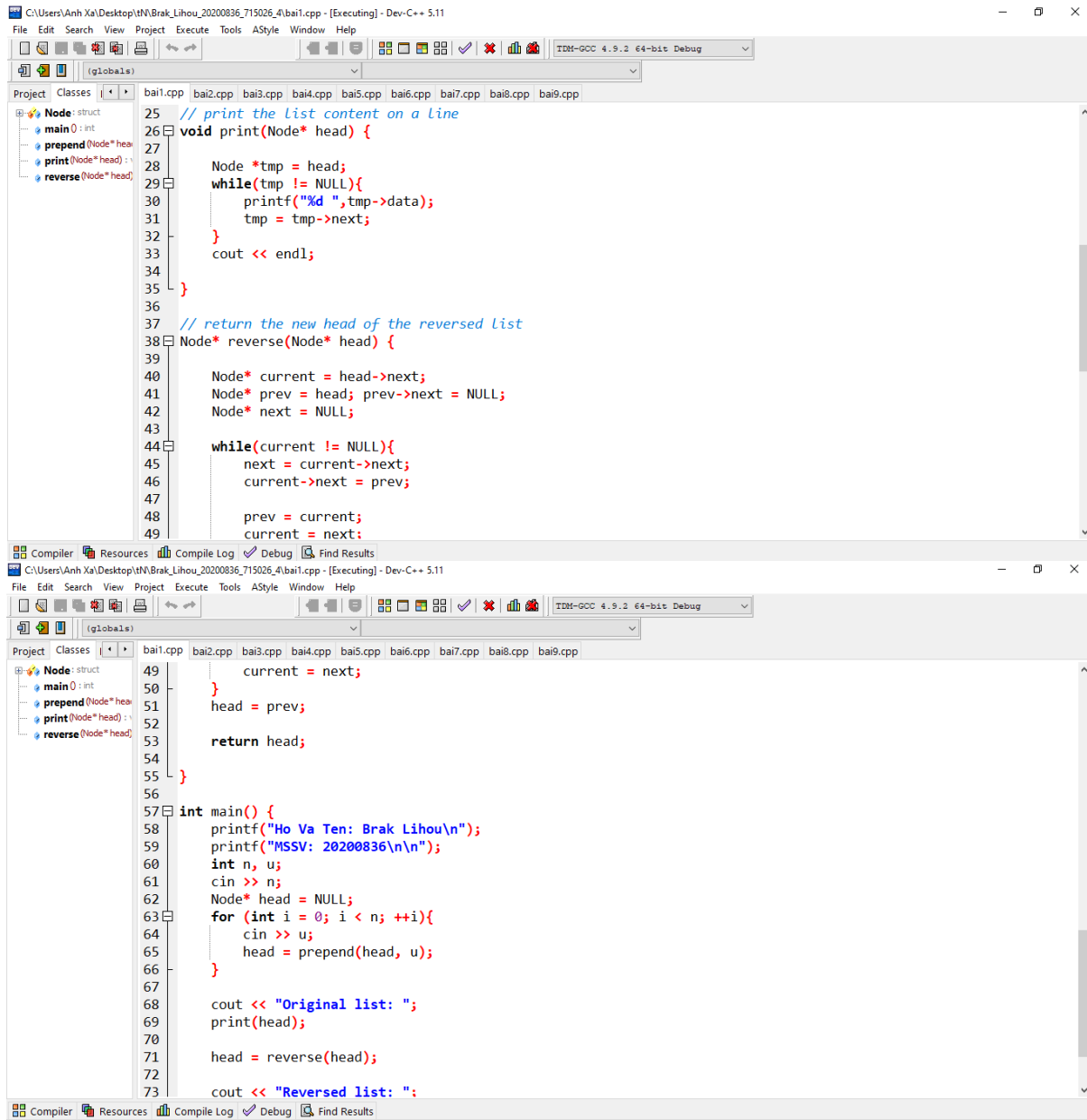
Check

The image shows a C++ IDE with the following components:

- Code Editor:** Contains C++ code for reversing a linked list. The code includes a `Node` struct, a `reverse` function, and a `main` function that reads input, reverses the list, and prints the result.
- Input/Output Table:**

Input	Result
10 -1 4 5 7 2 4 6 7 12 50	Original list: 50 12 7 6 4 2 7 5 4 -1 Reversed list: -1 4 5 7 2 4 6 7 12 50
- Output Window:** Displays the execution output, showing the original and reversed lists, and a message indicating the process exited after 26.61 seconds.

Báo Cao Thực hành buổi 4-Brak Lihou (20200836)



```
25 // print the list content on a line
26 void print(Node* head) {
27
28     Node *tmp = head;
29     while(tmp != NULL){
30         printf("%d ", tmp->data);
31         tmp = tmp->next;
32     }
33     cout << endl;
34 }
35
36
37 // return the new head of the reversed List
38 Node* reverse(Node* head) {
39
40     Node* current = head->next;
41     Node* prev = head; prev->next = NULL;
42     Node* next = NULL;
43
44     while(current != NULL){
45         next = current->next;
46         current->next = prev;
47
48         prev = current;
49         current = next;
50     }
51     head = prev;
52     return head;
53 }
54
55
56
57 int main() {
58     printf("Ho Va Ten: Brak Lihou\n");
59     printf("MSSV: 20200836\n\n");
60     int n, u;
61     cin >> n;
62     Node* head = NULL;
63     for (int i = 0; i < n; ++i){
64         cin >> u;
65         head = prepend(head, u);
66     }
67
68     cout << "Original list: ";
69     print(head);
70
71     head = reverse(head);
72
73     cout << "Reversed list: ";
```

4.2 Một điểm trong không gian 2 chiều được biểu diễn bằng pair. Hãy viết hàm tính diện tích tam giác theo tọa độ 3 đỉnh.

The screenshot displays a C++ IDE with the following components:

- Code Editor:** Contains the implementation of the `area` function. It includes headers for `<iostream>`, `<cmath>`, `<iomanip>`, and `<utility>`. It uses the `std::pair` type for points. The `distance` function calculates the Euclidean distance between two points. The `area` function calculates the area of a triangle using Heron's formula.
- Test Table:** A table with 4 columns: Test, Expected, Got, and a status column. It shows two test cases for the `area` function.
- Console Output:** A window showing the output of the program, which is the value 61.44.

Test	Expected	Got	
cout << setprecision(2) << fixed; cout << area({1, 2}, {2.5, 10}, {15, -5.25}) << endl;	61.44	61.44	✓
cout << setprecision(2) << fixed; cout << area({1, 2.5}, {2.5, 15}, {-5.2, -5.75}) << endl;	32.56	32.56	✓

```

1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 #include <utility>
5 using namespace std;
6 using Point = pair<double, double>;
7
8 double distance(Point a, Point b){
9     return sqrt((a.first - b.first)*(a.first - b.first) + (a.second - b.second)*(a.second - b.second));
10 }
11
12 double area(Point a, Point b, Point c) {
13     /******
14      * Ho va ten : Brak Lihou
15      * MSSV: 20200836
16      */
17     double ab = distance(a,b);
18     double bc = distance(b,c);
19     double ca = distance(c,a);
20
21     double p = (ab + bc + ca) /2;
22     return sqrt(p * (p-ab) * (p-bc) * (p-ca));
23     /******
24     */
25 }

```

Compiler: g++ 4.9.2 64-bit Debug

Project: bai1.cpp, bai2.cpp, bai3.cpp, bai4.cpp, bai5.cpp, bai6.cpp, bai7.cpp, bai8.cpp, bai9.cpp

Console Output:

```

MSSV: 20200836
61.44
Process exited after 0.1767 seconds with return value 0
Press any key to continue . . .

```

4.3 Một vector trong không gian 3 chiều được biểu diễn bằng tuple<double, double, double>. Hãy viết hàm tính tích có hướng của 2 vector.

Answer: (penalty regime: 10, 20, ... %)

```

1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 using namespace std;
5 using Vector = tuple<double, double, double>;
6
7 Vector cross_product(Vector a, Vector b) {
8     /*****
9      Ho va ten : Brak Lihou
10     MSSV : 20200836
11     */
12     double x = (get<1>(a)*get<2>(b) - get<1>(b)*get<2>(a));
13     double y = (get<2>(a)*get<0>(b) - get<0>(a)*get<2>(b));
14     double z = (get<0>(a)*get<1>(b) - get<0>(b)*get<1>(a));
15     return {x,y,z};
16     /*****/
17 }
18

```

Check

Test	Expected	Got
cout << setprecision(2) << fixed; Vector a {1.2, 4, -0.5}; Vector b {1.5, -2, 2.5};	9.00 -3.75 -8.40	9.00 -

C:\Users\Anh.Xa\Desktop\tnBrak_Lihou_20200836_715026_4\bai3.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes | bai1.cpp | bai2.cpp | bai3.cpp | bai4.cpp | bai5.cpp | bai6.cpp | bai7.cpp | bai8.cpp | bai9.cpp

```

1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 #include <vector>
5 using namespace std;
6 using Vector = tuple<double, double, double>;
7
8 Vector cross_product(Vector a , Vector b) {
9     /*****
10     Ho va ten : Brak Lihou
11     MSSV : 20200836
12     */
13     double x = (get<1>(a)*get<2>(b) - get<1>(b)*get<2>(a));
14     double y = (get<2>(a)*get<0>(b) - get<0>(a)*get<2>(b));
15     double z = (get<0>(a)*get<1>(b) - get<0>(b)*get<1>(a));
16     return {x,y,z};
17 }
18
19
20 int main() {
21     printf("Ho Va Ten: Brak Lihou\n");
22     printf("MSSV: 20200836\n");
23     cout << setprecision(2) << fixed;
24     Vector a {1.2, 4, -0.5};
25     Vector b {1.5, -2, 2.5};

```

Ho Va Ten: Brak Lihou
MSSV: 20200836
9.00 -3.75 -8.40
...Program finished with exit code 0
Press ENTER to exit console.

Compiler (24) Resources Compile Log Debug Find Results

Bài 4.4. Cho hai `std::vector`, hãy xóa hết các phần tử chẵn, sắp xếp giảm dần các số trong cả 2 vector và trộn lại thành một vector cũng được sắp xếp giảm dần.

My courses Theme colours This course 3 Brak Lihou

Question 4
Correct
Mark 10.00 out of 10.00
Flag question

Bài 4.4. Cho hai `std::vector`, hãy xóa hết các phần tử chẵn, sắp xếp giảm dần các số trong cả 2 vector và trộn lại thành một vector cũng được sắp xếp giảm dần.
For example:

Input	Result
5 6	Odd elements of a: 3 7 -5
2 3 6 7 -5	Odd elements of b: 13 5 9 35
13 5 2 4 9 35	Decreasingly sorted a: 7 3 -5
	Decreasingly sorted b: 35 13 9 5
	Decreasingly sorted c: 35 13 9 7 5 3 -5

Answer: (penalty regime: 10, 20, ... %)

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 void print_vector(const vector<int> &a) {
6     for (int v : a) cout << v << " ";
7     cout << endl;
8 }
9
10 bool compare(int a, int b){
11     return a > b;
12 }
13

```

courses Theme colours This course 3 Brak Lihou

Answer: (penalty regime: 10, 20, ... %)

```

11     return a > b;
12 }
13
14 vector<int> delete_even(vector<int> &a) {
15     vector<int> tmp;
16     for(int v : a){
17         if(v % 2 != 0) tmp.push_back(v);
18     }
19     return tmp;
20 }
21
22
23 void sort_decrease(vector<int> &a) {
24     sort(a.begin(), a.end(), compare);
25 }
26
27
28 vector<int> merge_vectors(const vector<int> &a, const vector<int> &b) {
29     //*****
30

```

Check

Theme colours This course 3 Brak Lihou

Answer: (penalty regime: 10, 20, ... %)

```

28 }
29 vector<int> merge_vectors(const vector<int> &a, const vector<int> &b) {
30     //*****
31     vector<int> sum;
32     for(int v : a){
33         sum.push_back(v);
34     }
35
36     for(int v : b){
37         sum.push_back(v);
38     }
39
40     sort(sum.begin(), sum.end(), compare);
41     return sum;
42 }
43
44
45 int main() {
46     //

```

Check

Theme colours This course Brak Lihou

Input	Result
5 6	Odd elements of a: 3 7 -5
2 3 6 7 -5	Odd elements of b: 13 5 9 35
13 5 2 4 9 35	Decreasingly sorted a: 7 3 -5
	Decreasingly sorted b: 35 13 9 5
	Decreasingly sorted c: 35 13 9 7 5 3 -5

Answer: (penalty regime: 10, 20, ... %)

```

44
45 int main() {
46
47     int m, n, u;
48     std::vector<int> a, b;
49
50     std::cin >> m >> n;
51     for(int i = 0; i < m; i++){
52         std::cin >> u;
53         a.push_back(u);
54     }
55     for(int i = 0; i < n; i++){
56         std::cin >> u;
57         b.push_back(u);
58     }

```

Theme colours This course Brak Lihou

Input	Expected
5 6	Odd elements of a: 3 7 -5
2 3 6 7 -5	Odd elements of b: 13 5 9 35
13 5 2 4 9 35	Decreasingly sorted a: 7 3 -5
	Decreasingly sorted b: 35 13 9 5
	Decreasingly sorted c: 35 13 9 7 5 3 -5

Answer: (penalty regime: 10, 20, ... %)

```

61     cout << "Odd elements of a: ";
62     print_vector(a);
63
64     b = delete_even(b);
65     cout << "Odd elements of b: ";
66     print_vector(b);
67
68     sort_decrease(a);
69     cout << "Decreasingly sorted a: ";
70     print_vector(a);
71
72     sort_decrease(b);
73     cout << "Decreasingly sorted b: ";
74     print_vector(b);
75
76     vector<int> c = merge_vectors(a, b);
77     cout << "Decreasingly sorted c: ";

```

Input	Expected
5 6	Odd elements of a: 3 7 -5
2 3 6 7 -5	Odd elements of b: 13 5 9 35
13 5 2 4 9 35	Decreasingly sorted a: 7 3 -5
	Decreasingly sorted b: 35 13 9 5
	Decreasingly sorted c: 35 13 9 7 5 3 -5

Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 void print_vector(const vector<int> &a) {
6     for (int v : a) cout << v << ' ';
7     cout << endl;
8 }
9
10 bool compare(int a, int b){
11     return a > b;
12 }
13
14 vector<int> delete_even(vector<int> &a) {
15
16     vector<int> tmp;
17     for(int v : a){
18         if(v % 2 != 0) tmp.push_back(v);
19     }
20     return tmp;
21 }
22
23
24 void sort_decrease(vector<int> &a) {
25     sort(a.begin(), a.end(), compare);

```

Compiler Resources Compile Log Debug Find Results

Select C:\Users\Anh Xa\Desktop\T\Brak_Lihou_20200836_715026_4\bai4.exe

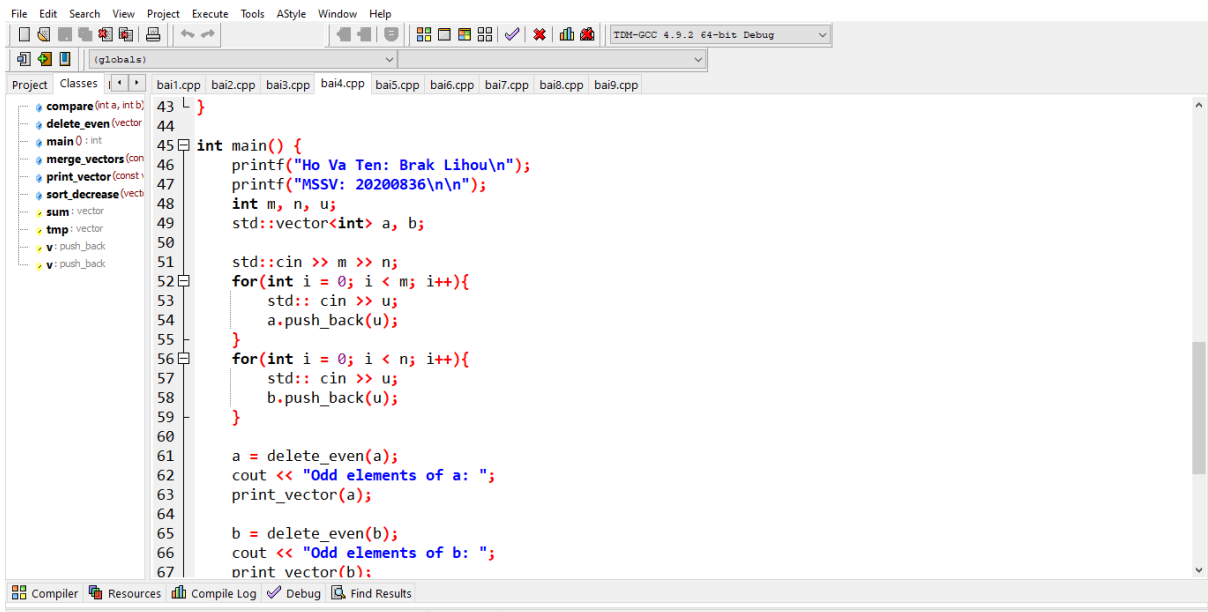
Ho Va Ten: Brak Lihou
HSSV: 20200836

```

5 6
2 3 6 7 -5
13 5 2 4 9 35
Odd elements of a: 3 7 -5
Odd elements of b: 13 5 9 35
Decreasingly sorted a: 7 3 -5
Decreasingly sorted b: 35 13 9 5
Decreasingly sorted c: 35 13 9 7 5 3 -5
.....
Process exited after 14.1 seconds with return value 0
Press any key to continue . . .

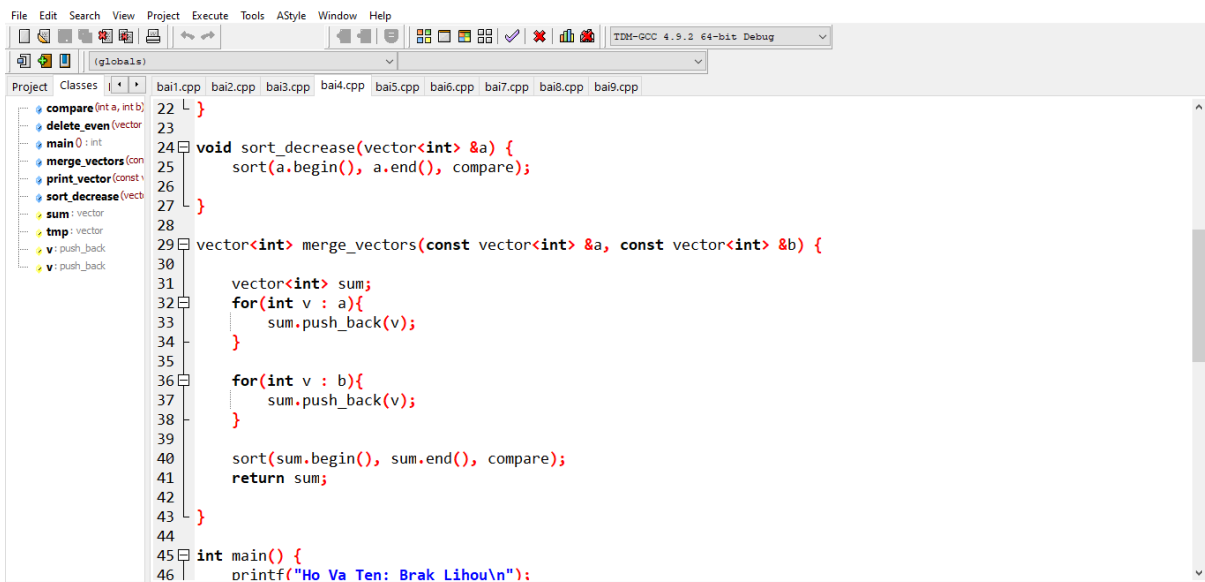
```


Báo Cao Thực hành buổi 4-Brak Lihou (20200836)



```
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp
compare(int a, int b)
delete_even(vector<int> &a)
main() : int
merge_vectors(const vector<int> &a, const vector<int> &b)
print_vector(const vector<int> &a)
sort_decrease(vector<int> &a)
sum : vector
tmp : vector
v : push_back
v : push_back

43 }
44
45 int main() {
46     printf("Ho Va Ten: Brak Lihou\n");
47     printf("MSSV: 20200836\n\n");
48     int m, n, u;
49     std::vector<int> a, b;
50
51     std::cin >> m >> n;
52     for(int i = 0; i < m; i++){
53         std::cin >> u;
54         a.push_back(u);
55     }
56     for(int i = 0; i < n; i++){
57         std::cin >> u;
58         b.push_back(u);
59     }
60
61     a = delete_even(a);
62     cout << "Odd elements of a: ";
63     print_vector(a);
64
65     b = delete_even(b);
66     cout << "Odd elements of b: ";
67     print_vector(b);
68 }
```



```
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp
compare(int a, int b)
delete_even(vector<int> &a)
main() : int
merge_vectors(const vector<int> &a, const vector<int> &b)
print_vector(const vector<int> &a)
sort_decrease(vector<int> &a)
sum : vector
tmp : vector
v : push_back
v : push_back

22 }
23
24 void sort_decrease(vector<int> &a) {
25     sort(a.begin(), a.end(), compare);
26 }
27
28
29 vector<int> merge_vectors(const vector<int> &a, const vector<int> &b) {
30
31     vector<int> sum;
32     for(int v : a){
33         sum.push_back(v);
34     }
35
36     for(int v : b){
37         sum.push_back(v);
38     }
39
40     sort(sum.begin(), sum.end(), compare);
41     return sum;
42 }
43
44
45 int main() {
46     printf("Ho Va Ten: Brak Lihou\n");
```

Bài 4.5. Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề. Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán DFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề. Yêu cầu hàm trả ra thứ tự các đỉnh được thăm (những đỉnh không thể thăm từ đỉnh 1 thì không phải in ra).

```

adj[6].push_back(7);
dfs(adj);

Answer: (penalty regime: 10, 20, ... %)

1 void dfs(std::vector< std::list<int> > adj) {
2     std::stack<int> S;
3     std::vector<bool> visited(adj.size());
4     S.push(1); // B?t d?u t? d?nh s? 1
5     while (!S.empty()) {
6         int u=S.top();
7         if (!visited[u]){
8             visited[u] = true;
9             std::cout<< u << std::endl;
10        }
11        if (!adj[u].empty()){
12            int v=adj[u].front(); adj[u].pop_front();
13            if (!visited[v]){
14                S.push(v);
15            }
16        } else { S.pop();}
17    }
18 }
19
20

Check

File Edit Search View Project Execute Tools AStyle Window Help
(globais)
Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp
dfs (std::vector< std::
main0: int
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 void dfs(std::vector< std::list<int> > adj) {
5     std::stack<int> S;
6     std::vector<bool> visited(adj.size());
7     S.push(1); // B?t d?u t? d?nh s? 1
8     while (!S.empty()) {
9         int u=S.top();
10        if (!visited[u]){
11            visited[u] = true;
12            std::cout<< u << std::endl;
13        }
14        if (!adj[u].empty()){
15            int v=adj[u].front(); adj[u].pop_front();
16            if (!visited[v]){
17                S.push(v);
18            }
19        } else { S.pop();}
20    }
21 }
22
23 int main(){
24     printf("Ho Va Ten: Brak Lihou\n");
25     printf("MSSV: 20200836\n");
26     int n = 7;
27     vector< list<int> > adj;
28     adj.resize(n + 1);
29     adj[1].push_back(2);
30     adj[2].push_back(4);
31     adj[1].push_back(3);
32     adj[3].push_back(4);
33     adj[3].push_back(5);
34     adj[5].push_back(2);
35     adj[2].push_back(7);
36     adj[6].push_back(7);
37     dfs(adj);
38 }

C:\Users\Anh Xa\Desktop\Brak_Lihou_20200836_715026_4\bai5.exe
Ho Va Ten: Brak Lihou
MSSV: 20200836
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
Process exited after 0.1099 seconds with return value 0
Press any key to continue . . .

```

Bài 4.6. Viết hàm void bfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy trên đồ thị biểu diễn bằng danh sách kề. Đồ thị có n đỉnh được đánh số từ 1 đến n. Thuật toán BFS xuất phát từ đỉnh 1. Các đỉnh được thăm theo thứ tự ưu tiên từ trái sang phải trong danh sách kề. Yêu cầu hàm trả

The screenshot shows a C++ IDE with two files open. The top file contains the BFS function implementation, and the bottom file contains the main function.

```

// bfs.cpp
void bfs(vector< list<int> > adj) {
    queue<int> Q;
    vector<bool> visited(adj.size());
    Q.push(1); // B?t d?u t? d?nh s? 1
    while (!Q.empty()) {
        int u=Q.front();
        if (!visited[u]){
            visited[u] = true;
            std::cout<< u << std::endl;
        }
        if (!adj[u].empty()){
            int v=adj[u].front(); adj[u].pop_front();
            if (!visited[v]){
                Q.push(v);
            }
        }
        else { Q.pop(); }
    }
    /*****/
}

// bai9.cpp
int v=adj[u].front(); adj[u].pop_front();
if (!visited[v]){
    Q.push(v);
}
} else { Q.pop(); }
}

int main(){
    printf("Ho Va Ten: Brak Lihou\n");
    printf("MSSV: 20200836\n");
    int n = 7;
    vector< list<int> > adj;
    adj.resize(n + 1);
    adj[1].push_back(2);
    adj[2].push_back(4);
    adj[1].push_back(3);
    adj[3].push_back(4);
    adj[3].push_back(5);
    adj[5].push_back(2);
    adj[2].push_back(7);
    adj[6].push_back(7);
    bfs(adj);
}
    
```

ra thứ tự các đỉnh được thăm (những đỉnh không thể thăm từ đỉnh 1 thì không phải in ra).

Bài 4.7. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set

BKICT Elearning English (en) My courses Theme colours This course 3 Brak Lihou

```

set<int> d = set_intersection(a, b);

cout << "Union: "; print_set(c);
cout << "Intersection: "; print_set(d);

```

Answer: (penalty regime: 10, 20, ... %)

```

1
2 template<class T>
3 set<T> set_union(const set<T> &a, const set<T> &b) {
4     set<T> c;
5     for(T v : a){
6         c.insert(v);
7     }
8
9     for(T v : b){
10        c.insert(v);
11    }
12
13    return c;
14    /*****/
15 }

```

BKICT Elearning English (en) My courses Theme colours This course 3 Brak Lihou

```

15 }
16
17 template<class T>
18 set<T> set_intersection(const set<T> &a, const set<T> &b) {
19     /*****/
20     set<T> c;
21     for(T v : a){
22         int index = 0;
23         for(T u : b){
24             if(v == u) index++;
25         }
26         if(index != 0) c.insert(v);
27     }
28     return c;
29     /*****/
30 }
31

```

Check

Test	Expected
✓ set<int> a = {1, 2, 3, 5, 7}; set<int> b = {2, 4, 5, 6, 9}; set<int> c = set_union(a, b);	Union: 1 2 3 4 5 6 7 9 Intersection: 2 5

Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp

```

1 #include <iostream>
2 #include <set>
3
4 using namespace std;
5
6 template<class T>
7 set<T> set_union(const set<T> &a, const set<T> &b) {
8     set<T> c;
9     for(T v : a){
10        c.insert(v);
11    }
12
13    for(T v : b){
14        c.insert(v);
15    }
16
17    return c;
18 }
19
20
21 template<class T>
22 set<T> set_intersection(const set<T> &a, const set<T> &b) {
23
24     set<T> c;
25     for(T v : a){

```

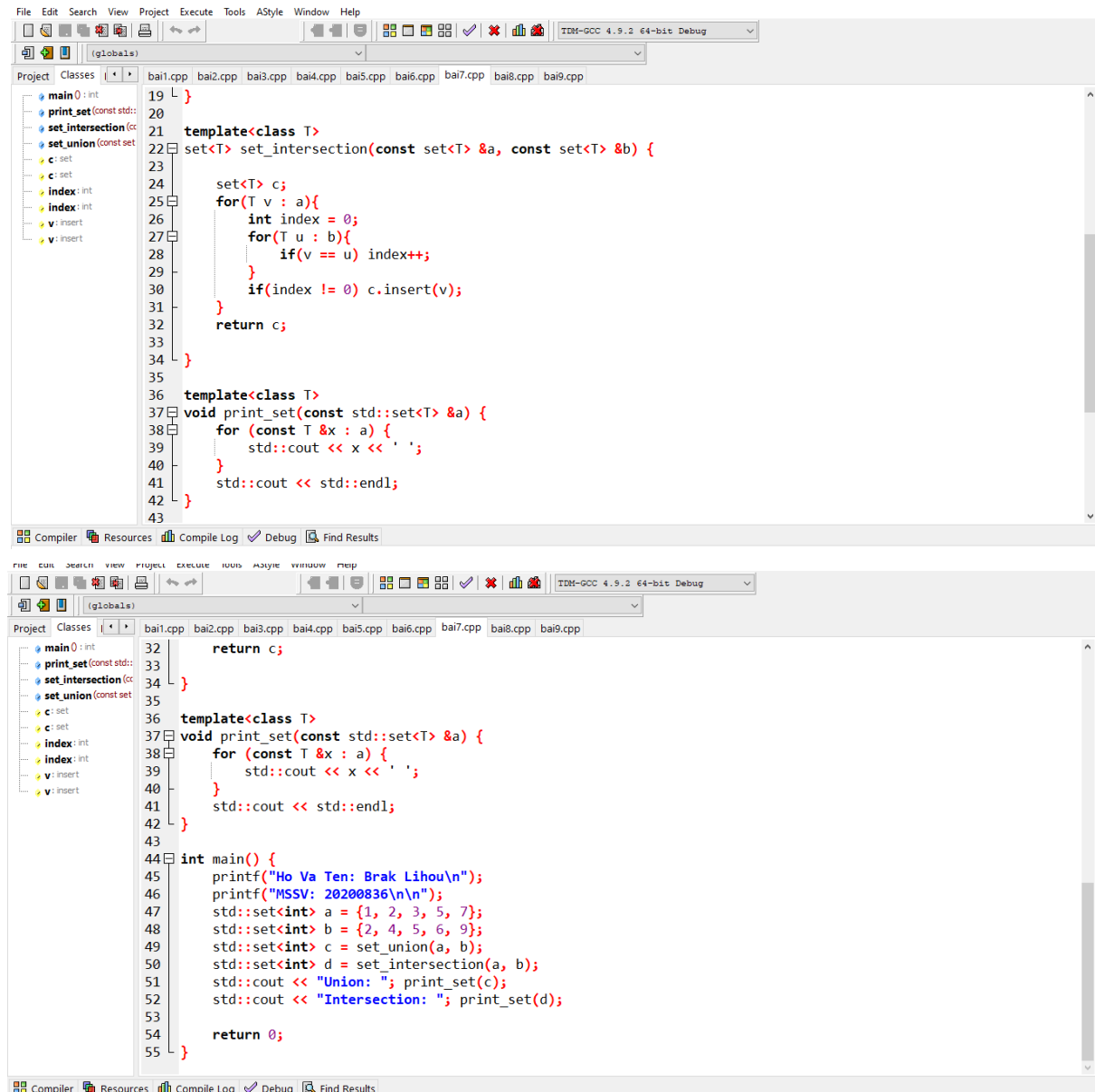
C:\Users\Anh.Xa\Desktop\tn\Brak_Lihou_20200836_715026_4\bai7.exe

Ho Va Ten: Brak Lihou
MSSV: 20200836

Union: 1 2 3 4 5 6 7 9
Intersection: 2 5

Process exited after 0.1206 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results



The image displays two screenshots of a C++ IDE, likely Visual Studio Code, showing the implementation of set operations using the STL.

Top Screenshot: Shows the implementation of a template function `set_intersection` and a template class `set`. The `set_intersection` function takes two constant references to `set<T>` and returns a new `set<T>` containing the intersection of the two sets. The `set` class has a `print_set` method that prints the elements of the set.

```

19 }
20
21 template<class T>
22 set<T> set_intersection(const set<T> &a, const set<T> &b) {
23
24     set<T> c;
25     for(T v : a){
26         int index = 0;
27         for(T u : b){
28             if(v == u) index++;
29         }
30         if(index != 0) c.insert(v);
31     }
32     return c;
33 }
34
35
36 template<class T>
37 void print_set(const std::set<T> &a) {
38     for (const T &x : a) {
39         std::cout << x << ' ';
40     }
41     std::cout << std::endl;
42 }
43

```

Bottom Screenshot: Shows the `main` function that demonstrates the use of the `set` class and the `set_intersection` function. It creates two sets, `a` and `b`, and calculates their union and intersection.

```

32     return c;
33 }
34
35
36 template<class T>
37 void print_set(const std::set<T> &a) {
38     for (const T &x : a) {
39         std::cout << x << ' ';
40     }
41     std::cout << std::endl;
42 }
43
44 int main() {
45     printf("Ho Va Ten: Brak Lihou\n");
46     printf("MSSV: 20200836\n\n");
47     std::set<int> a = {1, 2, 3, 5, 7};
48     std::set<int> b = {2, 4, 5, 6, 9};
49     std::set<int> c = set_union(a, b);
50     std::set<int> d = set_intersection(a, b);
51     std::cout << "Union: "; print_set(c);
52     std::cout << "Intersection: "; print_set(d);
53
54     return 0;
55 }

```

Bài 4.8. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.

BKICT Elearning English (en) My courses Theme colours This course 3 Brak Lihou

```

Answer: (penalty regime: 10, 20, ... %)
27     return c;
28 }
29
30
31 template<class T>
32 map<T, double> fuzzy_set_intersection(const map<T, double> &a, const map<T, double>
33 map<T, double> c;
34
35 for(auto v : a){
36     for(auto u : b){
37         if(v.first == u.first){
38             c.insert({v.first, min(v.second, u.second)});
39             break;
40         }
41     }
42 }
43
44 return c;
45 }
    
```

Check

Test

```

✓ map<int, double> a = {{1, 0.2}, {2, 0.5}, {3, 1}, {4, 0.6}, {5, 0.7}};
  map<int, double> b = {{1, 0.5}, {2, 0.4}, {4, 0.9}, {5, 0.4}, {6, 1}};
  cout << "A = "; print_fuzzy_set(a);
  cout << "B = "; print_fuzzy_set(b);
    
```

BKICT Elearning English (en) My courses Theme colours This course 3 Brak Lihou

```

map<int, double> d = fuzzy_set_intersection(a, b);
cout << "Union: "; print_fuzzy_set(c);
cout << "Intersection: "; print_fuzzy_set(d);
    
```

Answer: (penalty regime: 10, 20, ... %)

```

1  template<class T>
2  map<T, double> fuzzy_set_union(const map<T, double> &a, const map<T, double> &b)
3  map<T, double> c;
4  for(auto v : a){
5      int index = 0;
6      for(auto u : b){
7          if(v.first == u.first){
8              c.insert({v.first, max(v.second, u.second)});
9              index++;
10             break;
11         }
12     }
13     if(index == 0) c.insert({v.first, v.second});
14 }
15
16 for(auto v : b){
17     int index = 0;
18     for(auto u : a){
19         if(v.first == u.first){
20
    
```

Check

Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp

```

1  #include <iostream>
2  #include <map>
3  using namespace std;
4
5  template<class T>
6  map<T, double> fuzzy_set_union(const map<T, double> &a, const map<T, double> &b) {
7      map<T, double> c;
8      for(auto v : a){
9          int index = 0;
10         for(auto u : b){
11             if(v.first == u.first){
12                 c.insert({v.first, max(v.second, u.second)});
13                 index++;
14                 break;
15             }
16         }
17         if(index == 0) c.insert({v.first, v.second});
18     }
19
20     for(auto v : b){
21         int index = 0;
22         for(auto u : a){
23             if(v.first == u.first){
24                 c.insert({v.first, max(v.second, u.second)});
25                 index++;
    
```

Ho Va Ten: Brak Lihou
HSSV: 20200836

```

A = { (1, 0.2) (2, 0.5) (3, 1) (4, 0.6) (5, 0.7) }
B = { (1, 0.5) (2, 0.4) (4, 0.9) (5, 0.4) (6, 1) }
Union: { (1, 0.5) (2, 0.5) (3, 1) (4, 0.9) (5, 0.7) (6, 1) }
Intersection: { (1, 0.2) (2, 0.4) (4, 0.6) (5, 0.4) }

-----
Process exited after 0.2059 seconds with return value 0
Press any key to continue . . .
    
```

The image consists of two screenshots of a C++ IDE, likely Visual Studio Code, showing the implementation of fuzzy set operations. The top screenshot displays the `fuzzy_set_intersection` function, which takes two maps of integers to doubles and returns their intersection. The bottom screenshot displays the `fuzzy_set_union` function, which takes two maps and returns their union, and a `main` function that tests these operations with specific data sets.

```

25         index++;
26         break;
27     }
28     }
29     if(index == 0) c.insert({v.first, v.second});
30 }
31 return c;
32 }
33 }
34
35 template<class T>
36 map<T, double> fuzzy_set_intersection(const map<T, double> &a, const map<T, double> &b) {
37     map<T, double> c;
38
39     for(auto v : a){
40         for(auto u : b){
41             if(v.first == u.first){
42                 c.insert({v.first, min(v.second, u.second)});
43                 break;
44             }
45         }
46     }
47
48     return c;
49 }
50
51 template<class T>
52 void print_fuzzy_set(const std::map<T, double> &a) {
53     cout << "{";
54     for (const auto &x : a) {
55         std::cout << "(" << x.first << ", " << x.second << ") ";
56     }
57     cout << "}";
58     std::cout << std::endl;
59 }
60
61 int main() {
62     printf("Ho Va Ten: Brak Lihou\n");
63     printf("MSSV: 20200836\n");
64     std::map<int, double> a = {{1, 0.2}, {2, 0.5}, {3, 1}, {4, 0.6}, {5, 0.7}};
65     std::map<int, double> b = {{1, 0.5}, {2, 0.4}, {4, 0.9}, {5, 0.4}, {6, 1}};
66     std::cout << "A = "; print_fuzzy_set(a);
67     std::cout << "B = "; print_fuzzy_set(b);
68     std::map<int, double> c = fuzzy_set_union(a, b);
69     std::map<int, double> d = fuzzy_set_intersection(a, b);
70     std::cout << "Union: "; print_fuzzy_set(c);
71     std::cout << "Intersection: "; print_fuzzy_set(d);
72 }
73

```

Bài 4.9. Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh sách kề sử dụng std::priority_queue

BKICT Elearning
English (en)
My courses
Theme colours
This course
3
Brak Lihou

Answer: (penalty regime: 10, 20, ... %)

```

1 struct compare{
2     bool operator() (pair<int, int> a, pair<int, int> b){
3         return a.second > b.second;
4     }
5 };
6 vector<int> dijkstra(const vector< vector< pair<int, int> > > &adj) {
7     priority_queue<pair<int, int>, vector< pair<int, int> >, compare> q;
8
9     vector<int> dist(adj.size());
10    int dist_size = dist.size();
11    for(int i=0; i<dist_size; i++){
12        dist[i] = INT_MAX;
13    }
14    dist[0] = 0;
15    for(int i=0; i<dist_size; i++){
16        q.push({i, dist[i]});
17    }
18    while(!q.empty()){
19
20

```

Check

Test	Expected
✓ int n = 9; vector< vector< pair<int, int> > > adj(n);	distance 0->0 = 0 distance 0->1 = 4

BKICT Elearning English (en) My courses Theme colours This course Brak Lihou

```

18 }
19
20 while(!q.empty()){
21     pair<int, int> u_pair = q.top(); q.pop();
22     int u = u_pair.first;
23
24     for(auto v_pair : adj[u]){
25         int v = v_pair.first;
26         int weight = v_pair.second;
27
28         if(dist[v] > dist[u] + weight){
29             dist[v] = dist[u] + weight;
30             q.push({v, dist[v]});
31         }
32     }
33 }
34 return dist;
35 }

```

Check

Test	Expected
int n = 9;	distance 0->0 = 0
vector< vector< pair<int, int> > > adj(n);	distance 0->1 = 4
auto add_edge = [&adj] (int u, int v, int w) {	distance 0->2 = 12
adj[u].push_back({v, w});	distance 0->3 = 19
adj[v].push_back({u, w});	distance 0->4 = 21
};	distance 0->5 = 11
	distance 0->6 = 9
	distance 0->7 = 8
	distance 0->8 = 14

Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp

```

1 #include <iostream>
2 #include <queue>
3 #include <limits>
4 using namespace std;
5 struct compare{
6     bool operator() (pair<int, int> a, pair<int, int> b){
7         return a.second > b.second;
8     }
9 };
10 vector<int> dijkstra(const vector< vector< pair<int, int> > > adj) {
11     priority_queue<pair<int, int>, vector< pair<int, int> >, compare> q;
12
13     vector<int> dist(adj.size());
14     int dist_size = dist.size();
15     for(int i=0; i<dist_size; i++){
16         dist[i] = INT_MAX;
17     }
18     dist[0] = 0;
19     for(int i=0; i<dist_size; i++){
20         q.push({i, dist[i]});
21     }
22
23     while(!q.empty()){
24         pair<int, int> u_pair = q.top(); q.pop();
25
26         pair<int, int> u_pair = q.top(); q.pop();
27         int u = u_pair.first;
28
29         for(auto v_pair : adj[u]){
30             int v = v_pair.first;
31             int weight = v_pair.second;
32
33             if(dist[v] > dist[u] + weight){
34                 dist[v] = dist[u] + weight;
35                 q.push({v, dist[v]});
36             }
37         }
38     }
39     return dist;
40 }
41 int main() {
42     int n = 9;
43     vector< vector< pair<int, int> > > adj(n);
44     auto add_edge = [&adj] (int u, int v, int w) {
45         adj[u].push_back({v, w});
46         adj[v].push_back({u, w});
47     };
48     add_edge(0, 1, 4);
49     add_edge(0, 7, 8);

```

C:\Users\Anh Xa\Desktop\tn\Brak_Lihou_20200836_715026_4\bai9... distance 0->0 = 0 distance 0->1 = 4 distance 0->2 = 12 distance 0->3 = 19 distance 0->4 = 21 distance 0->5 = 11 distance 0->6 = 9 distance 0->7 = 8 distance 0->8 = 14 Process exited after 0.1804 seconds with return value 0 Press any key to continue . . .

Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp

```

25 pair<int, int> u_pair = q.top(); q.pop();
26 int u = u_pair.first;
27
28 for(auto v_pair : adj[u]){
29     int v = v_pair.first;
30     int weight = v_pair.second;
31
32     if(dist[v] > dist[u] + weight){
33         dist[v] = dist[u] + weight;
34         q.push({v, dist[v]});
35     }
36 }
37 }
38 return dist;
39 }
40 int main() {
41     int n = 9;
42     vector< vector< pair<int, int> > > adj(n);
43     auto add_edge = [&adj] (int u, int v, int w) {
44         adj[u].push_back({v, w});
45         adj[v].push_back({u, w});
46     };
47
48     add_edge(0, 1, 4);
49     add_edge(0, 7, 8);

```

Compiler Resources Compile Log Debug Find Results

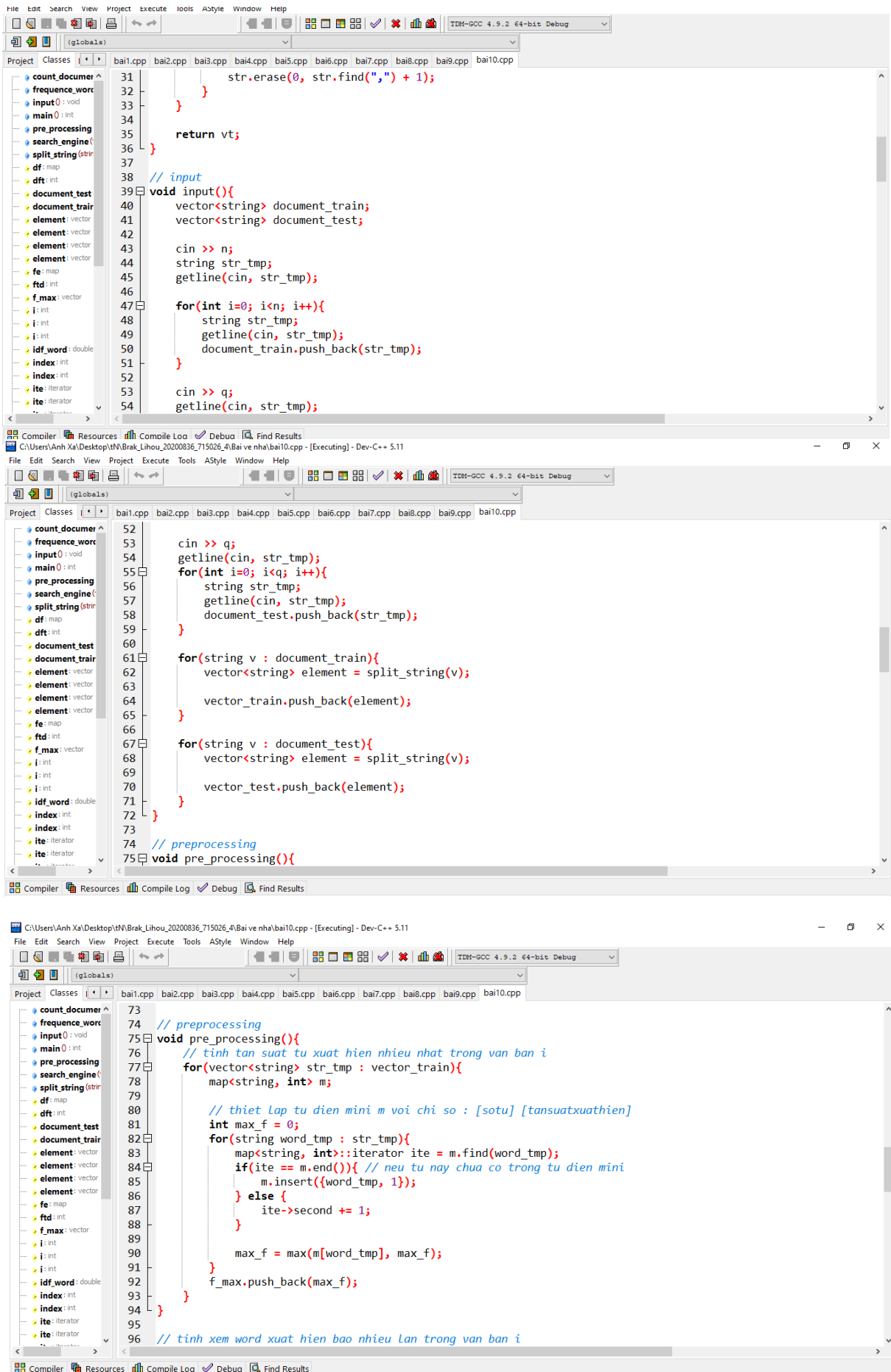
```

45         adj[v].push_back({u, w});
46     };
47
48     add_edge(0, 1, 4);
49     add_edge(0, 7, 8);
50     add_edge(1, 7, 11);
51     add_edge(1, 2, 8);
52     add_edge(2, 3, 7);
53     add_edge(2, 8, 2);
54     add_edge(3, 4, 9);
55     add_edge(3, 5, 14);
56     add_edge(4, 5, 10);
57     add_edge(5, 6, 2);
58     add_edge(6, 7, 1);
59     add_edge(6, 8, 6);
60     add_edge(7, 8, 7);
61
62     vector<int> distance = dijkstra(adj);
63     for (int i = 0; i < distance.size(); ++i) {
64         cout << "distance " << i << "->" << i << " = " << distance[i] << endl;
65     }
66
67     return 0;
68 }
    
```

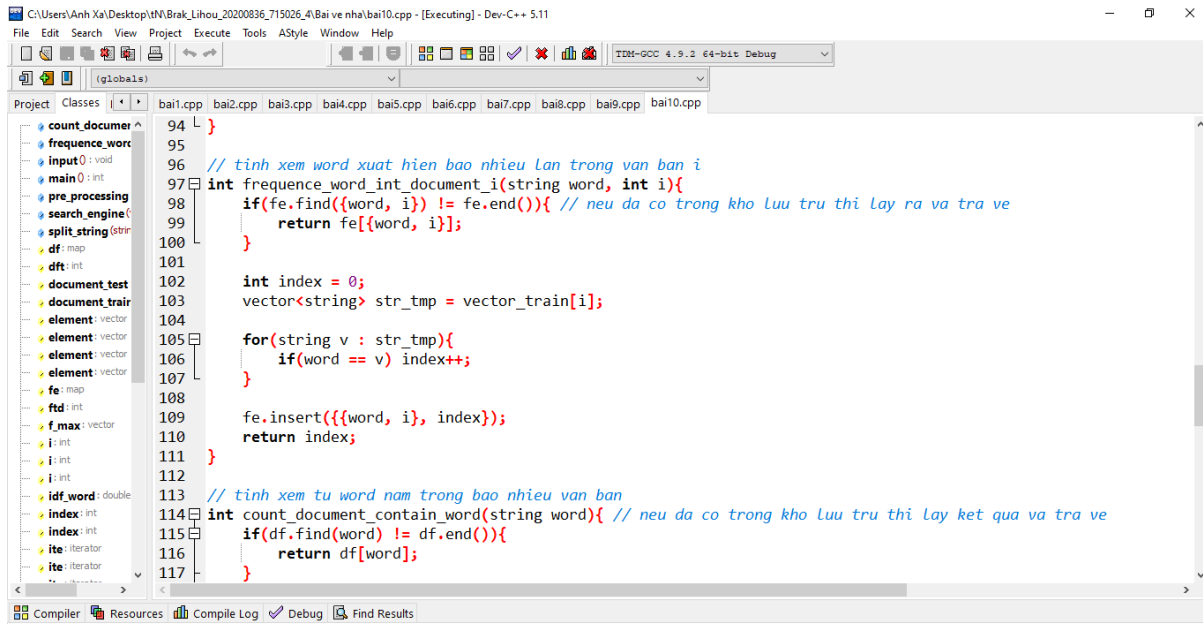
Bài tập 10: Search Engine

```

1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int n, q;
5 vector< vector<string> > vector_train;
6 vector< vector<string> > vector_test;
7 vector<int> f_max; // tan suat xuat hien tu xuat hi
8 map<string, int> df; // tinh xem tu word nam trong
9 map<pair<string, int>, int> fe; // tinh xem tu word
10
11 // split string
12 vector<string> split_string(string str){
13     vector<string> vt;
14
15     while (!str.empty()){
16         string tmp = str.substr(0, str.find(", "));
17         int pos = tmp.find(" ");
18
19         if(pos > tmp.size())
20             vt.push_back(tmp);
21         else {
22             while(pos <= tmp.size()){
23                 tmp.erase(pos, 1);
24                 pos = tmp.find(" ");
25             }
26             vt.push_back(tmp);
27         }
28         if (str.find(", " > str.size())
29             break;
30         else {
31             str.erase(0, str.find(", " + 1);
32         }
33     }
34 }
    
```

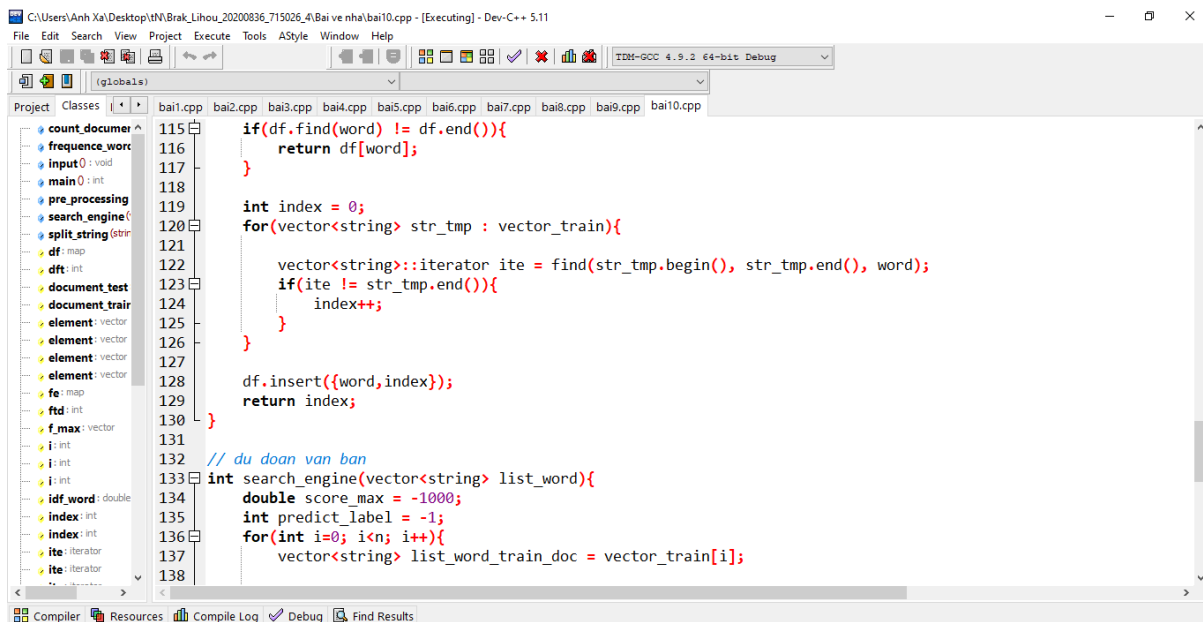


Báo Cao Thực hành buổi 4-Brak Lihou (20200836)



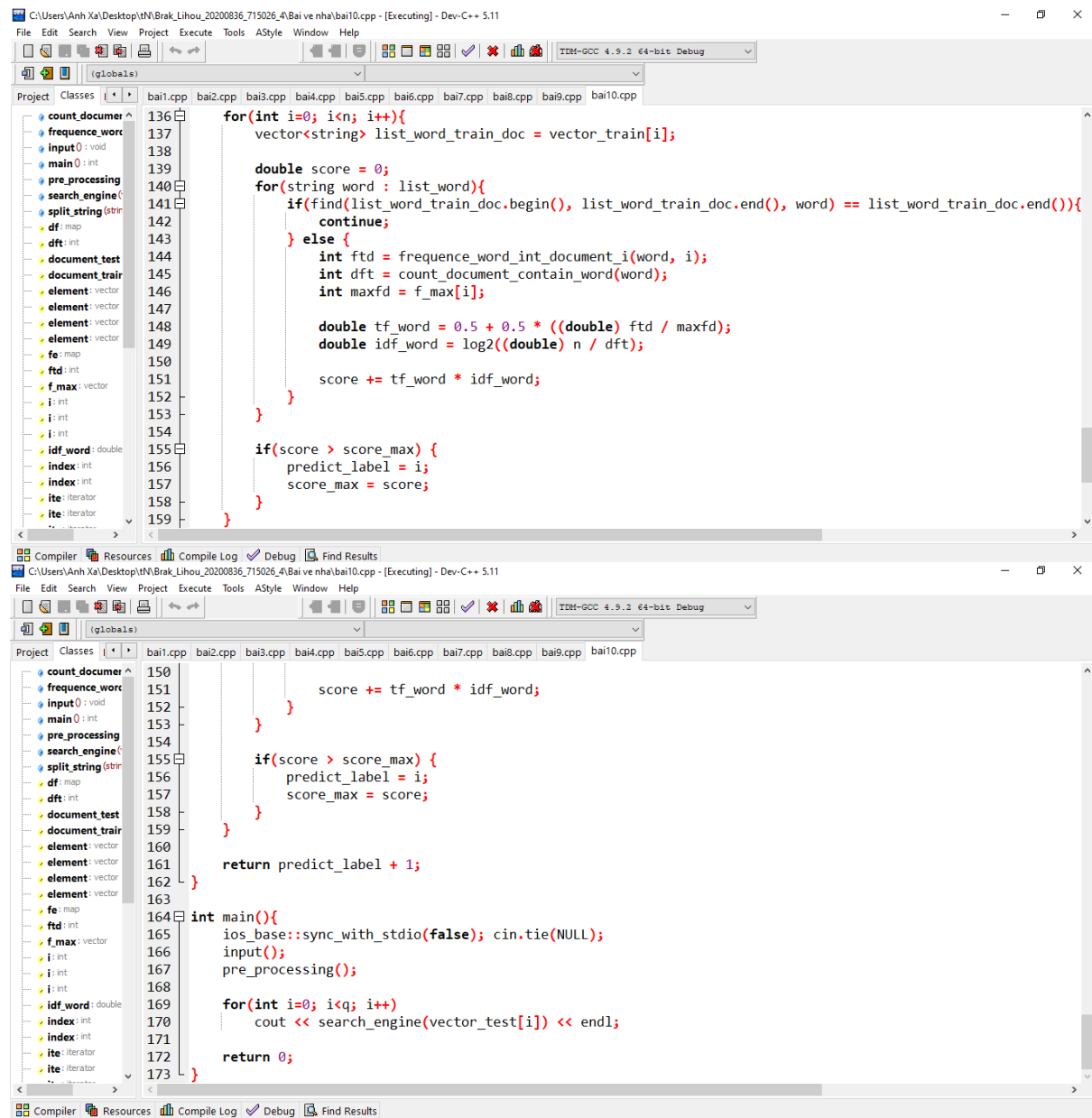
The screenshot shows a C++ IDE with the file `bai10.cpp` open. The code implements a function `frequency_word_int_document_i` that takes a string `word` and an integer `i` as input. It checks if the word is already in a frequency map `fe`. If not, it iterates through a vector of strings `str_tmp` (which is `vector_train[i]`) and increments the index for each occurrence of the word. Finally, it inserts the word and its index into the frequency map `fe` and returns the index.

```
94 }
95
96 // tinh xem word xuất hiện bao nhiêu lần trong văn bản i
97 int frequency_word_int_document_i(string word, int i){
98     if(fe.find({word, i}) != fe.end()){ // nếu đã có trong kho lưu trữ thì lấy ra và trả về
99         return fe[{word, i}];
100     }
101
102     int index = 0;
103     vector<string> str_tmp = vector_train[i];
104
105     for(string v : str_tmp){
106         if(word == v) index++;
107     }
108
109     fe.insert({{word, i}, index});
110     return index;
111 }
112
113 // tinh xem từ word nằm trong bao nhiêu văn bản
114 int count_document_contain_word(string word){ // nếu đã có trong kho lưu trữ thì lấy kết quả và trả về
115     if(df.find(word) != df.end()){
116         return df[word];
117     }
118 }
```

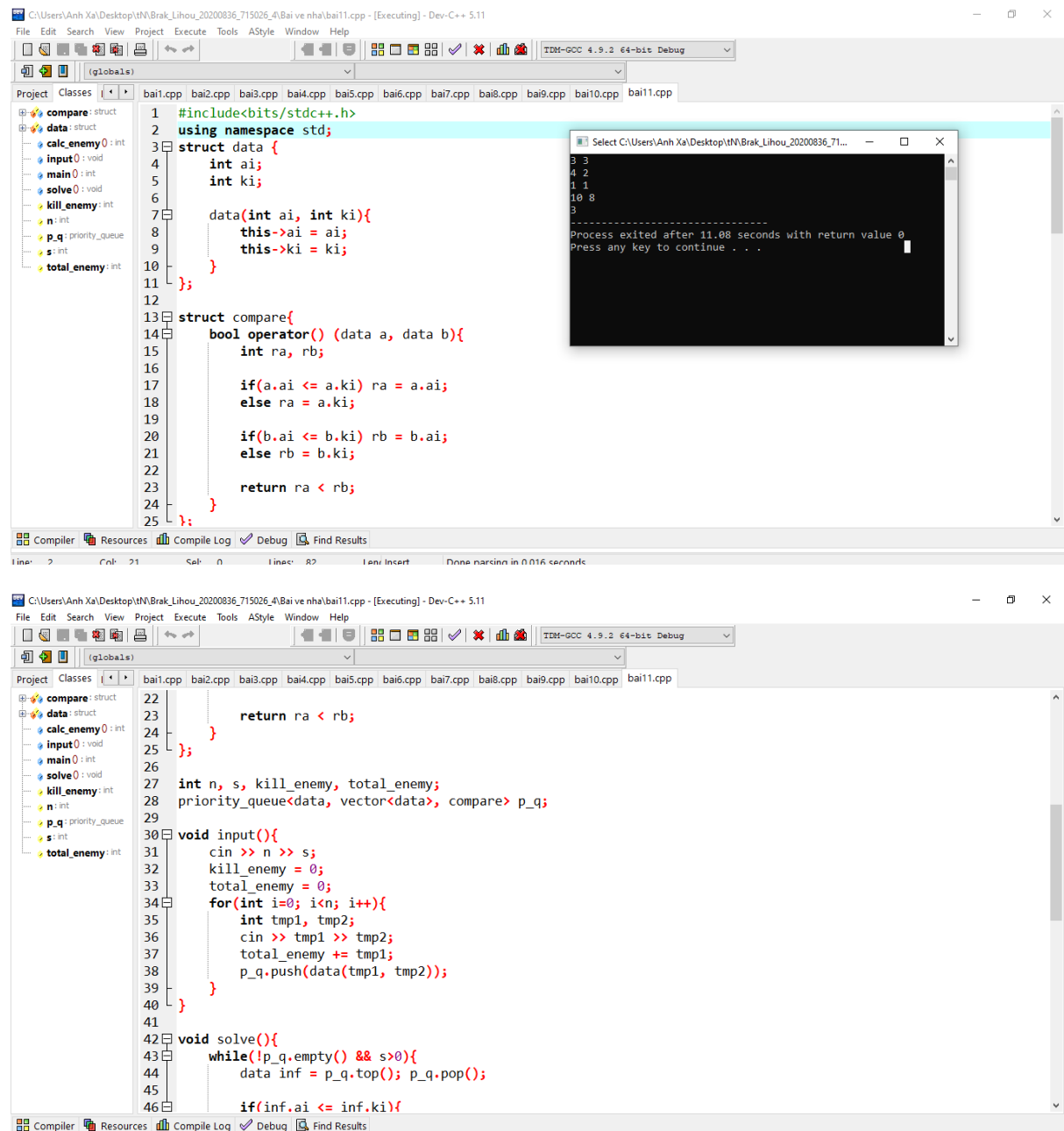


The screenshot shows a C++ IDE with the file `bai10.cpp` open. The code implements a function `search_engine` that takes a vector of strings `list_word` as input. It iterates through each word in the list and checks if it is already in a document frequency map `df`. If not, it iterates through a vector of strings `str_tmp` (which is `vector_train[i]`) and increments the index for each occurrence of the word. Finally, it inserts the word and its index into the frequency map `df` and returns the index.

```
115 if(df.find(word) != df.end()){
116     return df[word];
117 }
118
119 int index = 0;
120 for(vector<string> str_tmp : vector_train){
121     vector<string>::iterator ite = find(str_tmp.begin(), str_tmp.end(), word);
122     if(ite != str_tmp.end()){
123         index++;
124     }
125 }
126
127 df.insert({{word, index}});
128 return index;
129 }
130
131 // du đoán văn bản
132 int search_engine(vector<string> list_word){
133     double score_max = -1000;
134     int predict_label = -1;
135     for(int i=0; i<n; i++){
136         vector<string> list_word_train_doc = vector_train[i];
137     }
138 }
```



Báo Cao Thực hành buổi 4-Brak Lihou (20200836)



The image displays two screenshots of a C++ development environment (Dev-C++ 5.11) showing the implementation of a priority queue-based solution for a game simulation. The code is organized into multiple source files (bai1.cpp to bai11.cpp).

Top Screenshot: Shows the implementation of the `data` struct and the `compare` struct. The `data` struct contains `ai` and `ki` attributes. The `compare` struct defines a `operator()` function for sorting. A console window shows the execution output, indicating the process exited after 11.08 seconds with a return value of 0.

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 struct data {
4     int ai;
5     int ki;
6
7     data(int ai, int ki){
8         this->ai = ai;
9         this->ki = ki;
10    }
11 };
12
13 struct compare{
14     bool operator() (data a, data b){
15         int ra, rb;
16
17         if(a.ai <= a.ki) ra = a.ai;
18         else ra = a.ki;
19
20         if(b.ai <= b.ki) rb = b.ai;
21         else rb = b.ki;
22
23         return ra < rb;
24     }
25 };
```

Bottom Screenshot: Shows the implementation of the `input` and `solve` functions. The `input` function reads the number of enemies (`n`) and the number of soldiers (`s`), and initializes the `priority_queue` with the `data` struct. The `solve` function processes the queue until it is empty or the number of soldiers is zero.

```
22     return ra < rb;
23 }
24 }
25 };
26
27 int n, s, kill_enemy, total_enemy;
28 priority_queue<data, vector<data>, compare> p_q;
29
30 void input(){
31     cin >> n >> s;
32     kill_enemy = 0;
33     total_enemy = 0;
34     for(int i=0; i<n; i++){
35         int tmp1, tmp2;
36         cin >> tmp1 >> tmp2;
37         total_enemy += tmp1;
38         p_q.push(data(tmp1, tmp2));
39     }
40 }
41
42 void solve(){
43     while(!p_q.empty() && s>0){
44         data inf = p_q.top(); p_q.pop();
45         if(inf.ai <= inf.ki){
46             // ... (code continues) ...
```

Báo Cao Thực hành buổi 4-Brak Lihou (20200836)

```
C:\Users\Anh.Xa\Desktop\tn\Brak_Lihou_20200836_715026_4\ Bai ve nha\ bai11.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai8.cpp bai9.cpp bai10.cpp bai11.cpp
compare: struct
data: struct
calc_enemy0: int
input0: void
main0: int
solve0: void
kill_enemy: int
n: int
p.q: priority_queue
s: int
total_enemy: int
43 while(!p.q.empty() && s>0){
44     data inf = p.q.top(); p.q.pop();
45
46     if(inf.ai <= inf.ki){
47         kill_enemy += inf.ai;
48     } else {
49         int nenemy = inf.ai - inf.ki;
50         p.q.push(data(nenemy, inf.ki));
51         kill_enemy += inf.ki;
52     }
53
54     s -= 1;
55 }
56
57
58 int calc_enemy(){
59     /*
60     int sum = 0;
61     while(!p.q.empty()){
62         data v = p.q.top(); p.q.pop();
63         sum += v.ai;
64     }
65     return sum;
66     */
67     return total_enemy - kill_enemy;
68 }
69
70 int main(){
71     input();
72     solve();
73     cout << calc_enemy();
74
75     /*
76     while(!p.q.empty()){
77         auto v = p.q.top(); p.q.pop();
78         cout << v.ai << " " << v.ki << endl;
79     }
80     */
81     return 0;
82 }
```

Bài tập 12. Lược đồ

```

1 #include<bits/stdc++.h>
2 using namespace std;
3
4 long long n;
5 vector<long long> vt;
6 vector<long long> L; // mang trái lưu trữ giá trị
7 vector<long long> R; // mang phải lưu trữ giá trị
8
9 void inputData(){
10     cin >> n;
11     vt.push_back(-1); // Dữ liệu cạnh o 2 đầu mang
12     for(long long i=0; i<n; i++){
13         long long tmp;
14         cin >> tmp;
15         vt.push_back(tmp);
16     }
17     vt.push_back(-1); // Dữ liệu cạnh o 2 đầu mang
18 }
19
20 void solve(){
21     while(true){
22         inputData();
23         if(n==0) return;
24     }
25     // Xu Ly o day

```

```

25 // Xu Ly o day
26 stack<long long> st; // 1 stack để thực hiện thuật toán ghi ra các giá trị L, R
27 L.resize(n+2);
28 R.resize(n+2);
29 for(long long i=1; i<=n+1; i++){
30     while(!st.empty() && vt[i] < vt[st.top()]){
31         L[st.top()] = i;
32         st.pop();
33     }
34     st.push(i);
35 }
36
37 for(long long i=n; i>=0; i--){
38     while(!st.empty() && vt[i] < vt[st.top()]){
39         R[st.top()] = i;
40         st.pop();
41     }
42     st.push(i);
43 }
44
45 long long maxh = 0;
46 for(long long i=1; i<=n; i++){
47     long long h = (L[i] - R[i] - 1)*vt[i];
48     if(h > maxh) maxh = h;
49 }

```

```

40     st.pop();
41 }
42     st.push(i);
43 }
44
45 long long maxh = 0;
46 for(long long i=1; i<=n; i++){
47     long long h = (L[i] - R[i] - 1)*vt[i];
48     if(h > maxh) maxh = h;
49 }
50
51 cout << maxh << endl; break;
52
53 // Xóa mảng trước khi bắt đầu vòng lặp mới ( mảng vt, mảng L, mảng R)
54 vt.erase(vt.begin(), vt.end());
55 L.erase(L.begin(), L.end());
56 R.erase(R.begin(), R.end());
57 }
58
59
60 int main(){
61     solve();
62     return 0;
63 }

```


Bài tập 13: Đếm xâu con

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int n;
4 string str;
5 int cnt;
6 int res;
7 map<int, int> m;
8
9 void input(){
10     cin >> str;
11     n = str.length();
12     cnt = 0;
13     res = 0;
14 }
15
16 int calc(int b){
17     int foo = 0;
18     for(int i=1; i<b; i++){
19         foo += i;
20     }
21     return foo;
22 }
23
24 void solve(){
25     int slg0 = 0, slg1 = 0;
26     map<int, int>::iterator it;

```

Console Output:

```

1001011
8
Process exited after 8.723 seconds with return value 0
Press any key to continue . . .

```

```

27
28 int slg0 = 0, slg1 = 0;
29 map<int, int>::iterator it;
30
31 for(int i=0; i<n; i++){
32     if(str[i] == '0') slg0++;
33     else slg1++;
34
35     int tmp = slg1 - slg0; // do chenh Lenh 0,1
36     it = m.find(tmp);
37     if(it != m.end()){ // neu do chenh Lech da ton tai
38         it->second += 1;
39     } else {
40         m.insert({tmp, 1});
41     }
42 }
43
44 it = m.begin();
45 while(it != m.end()){
46     res += calc(it->second);
47     if(it->first == 0) res += it->second;
48     it++;
49 }

```

```

50
51 int tmp = slg1 - slg0; // do chenh Lenh 0,1
52 it = m.find(tmp);
53 if(it != m.end()){ // neu do chenh Lech da ton tai
54     it->second += 1;
55 } else {
56     m.insert({tmp, 1});
57 }
58
59 it = m.begin();
60 while(it != m.end()){
61     res += calc(it->second);
62     if(it->first == 0) res += it->second;
63     it++;
64 }
65
66 int main(){
67     input();
68     solve();
69
70     cout << res;
71     return 0;
72 }

```