Contents

[Danh mục hình ảnh 1](#__RefHeading___Toc349_2734074527)

[Bài thực hành số 5: Thực hành gỡ rối, kiềm thử và tinh chỉnh mã nguồn 3](#__RefHeading___Toc1901_2139985059)

[Phần 1. Thực hành gỡ rối 3](#__RefHeading___Toc353_2734074527)

[1. Bài tập 1: Tìm và sửa lỗi cú pháp 3](#__RefHeading___Toc355_2734074527)

[2. Bài tập 2: Tìm và sửa lỗi cú pháp 4](#__RefHeading___Toc357_2734074527)

[3. Bài tập 3: Dãy ngoặc đúng 6](#__RefHeading___Toc359_2734074527)

[4. Bài tập 4: Bài toán người du lịch 8](#__RefHeading___Toc2013_1126995683)

[5. Bài tập 5: Năm nhuận 10](#__RefHeading___Toc2223_442281320)

[6. Bài tập 6: Tổng kết 11](#__RefHeading___Toc2225_442281320)

[7. Bài tập 7: Chia tiền 12](#__RefHeading___Toc2227_442281320)

[8. Bài tập 8: Cắt hình chữ nhật 14](#__RefHeading___Toc2229_442281320)

# Danh mục hình ảnh

[Hình 1 Bài 1.1 Tìm và sửa lỗi cú pháp 4](#Hình!0|sequence)

[Hình 2 Bài 1.2 Tìm và sửa lỗi cú pháp 6](#Hình!1|sequence)

[Hình 3 Bài 1.3 Dãy ngoặc đúng 8](#Hình!2|sequence)

[Hình 4 Bài 1.4 Bài toán người du lịch 10](#Hình!3|sequence)

[Hình 5 Bài 1.5 Năm nhuận 11](#Hình!4|sequence)

[Hình 6 Bài 1.6 Tổng kết 12](#Hình!5|sequence)

[Hình 7 Bài 1.7 Chia tiền 14](#Hình!6|sequence)

[Hình 8 Bài 1.8 Cắt hình chữ nhật 15](#Hình!7|sequence)

# Bài thực hành số 5**: Thực hành gỡ rối, kiềm thử và tinh chỉnh mã nguồn**

## Phần 1. **Thực hành gỡ rối**

### Bài tập 1: **Tìm và sửa lỗi cú pháp**

Đoạn code sau liệt kê tất cả các hoán vị n số. Hãy tìm và sửa các lỗi cú pháp như hướng dẫn ở trên.

#include <stdio.h>

int x[100], mark[100], n;

void printf(){ // Wrong

for (int i = 1; i <= n; ++i) printf("%d ", x[i]);

printf("\n");

}

void process(int i) {

if (i > n){

printf(); // Error

return;

}

for (int j = 1; j <= n; ++j)

if (!mark[j]){

mark[j] = 1;

x[i] = j;

process(i+1);

mark[j] = 0;

}

}

int main() {

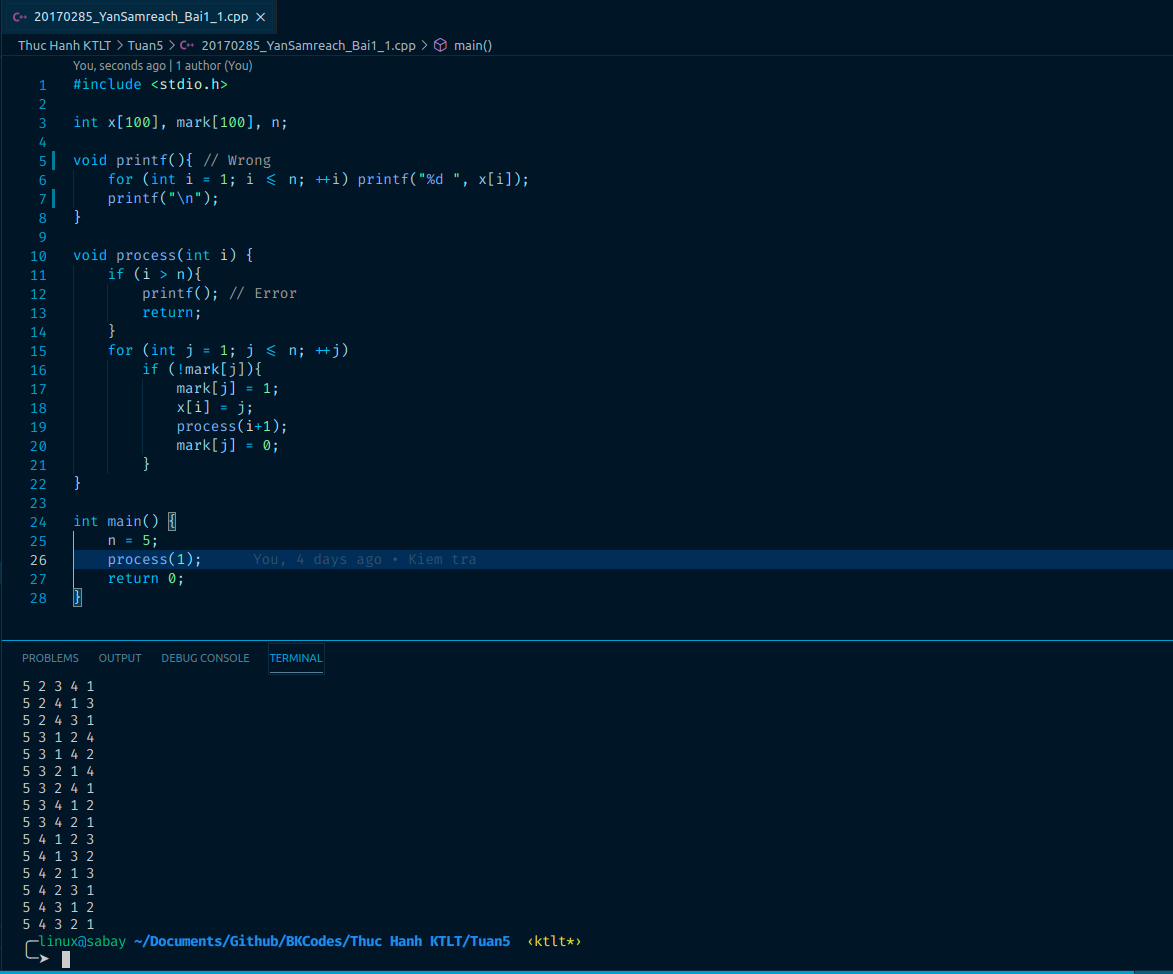
n = 5;

process(1);

return 0;

}

Kết quả:



Hình 1 Bài 1.1 Tìm và sửa lỗi cú pháp

### Bài tập 2: **Tìm và sửa lỗi cú pháp**

Bài toán cái túi: Cho một cái túi có sức chứa M và n đồ vật. Đồ vật thứ i có khối lượng mi và giá trị vi. Cần chọn ra một số đồ vật để bỏ vào túi sao cho tổng khối lượng không quá M và tổng giá trị là lớn nhất có thể. Đoạn code sau đây giải bài toán cái túi bằng phương pháp duyệt nhánh cận. Hãy tìm và sửa các lỗi cú pháp

#include <iostream>

using namespace std;

int n, M, m[100], v[100];

int x[100], best, sumV, sumM, all[100]; // Đổi tên All thành all

void init(){

for (int i = n; i >= 1; --i){

all[i] = all[i+1] + v[i];

}

}

void print() {

cout << best;

}

int process(int i){ // Doi kieu function

if (sumV + all[i] <= best || sumM > M) return 0; // lỗi cú pháp

if (i > n){

best = sumV; // Thiếu dấu chấm phẩy ;

return 0;

}

process(i+1);

sumM += m[i];

sumV += v[i];

process(i+1);

sumM -= m[i];

sumV -= v[i];

}

int main() {

cin >> n >> M;

for (int i = 1; i <= n; ++i)

cin >> m[i] >> v[i];

init();

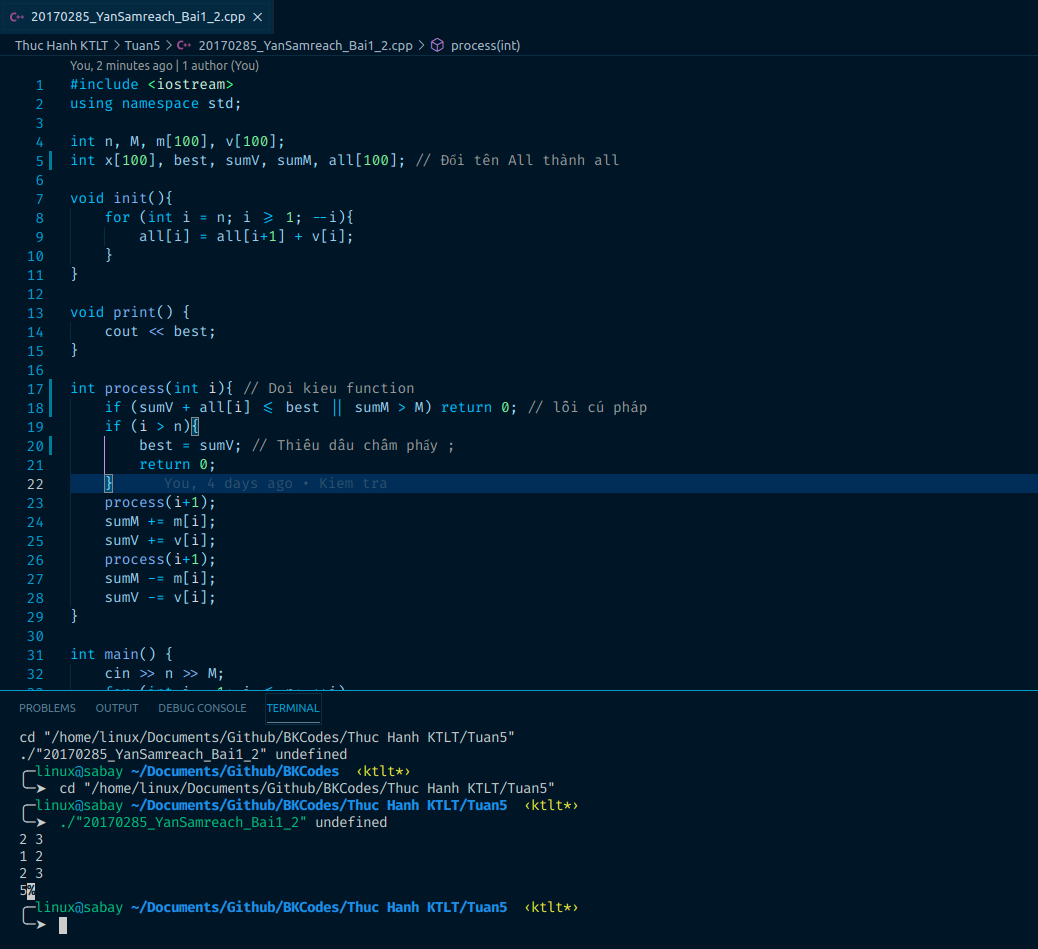
process(1);

print();

return 0;

}

Kết quả:



Hình 2 Bài 1.2 Tìm và sửa lỗi cú pháp

### Bài tập 3: Dãy ngoặc đúng

Đề bài: <http://codeforces.com/group/Ir5CI6f3FD/contest/269186/problem/H>

Mã nguồn dưới đây là của một sinh viên, khi submit bị lỗi runtime (Exit code is -1073741819). Sử dụng công cụ debug ở trên, hãy tìm và sửa các lỗi trong mã nguồn

#include <iostream>

using namespace std;

#include <string.h>

#include <stack>

int par(string str) {

int a = str.length();

stack<char> S;

char x, y;

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

x = str[i];

if (x == '(' || x == '[' || x == '{') {

S.push(x);

} else {

if (S.empty()) return 0;

if (x == ')') {

if (S.top() == '(') {

S.pop();

}

else return 0;

}

else if (x == ']') {

if (S.top() == '[') {

S.pop();

}

else return 0;

}

else if (x == '}') {

if (S.top() == '{') {

S.pop();

}

else return 0;

}

}

}

return S.empty();

}

int main() {

int n;

string str;

cin >> n;

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

cin >> str;

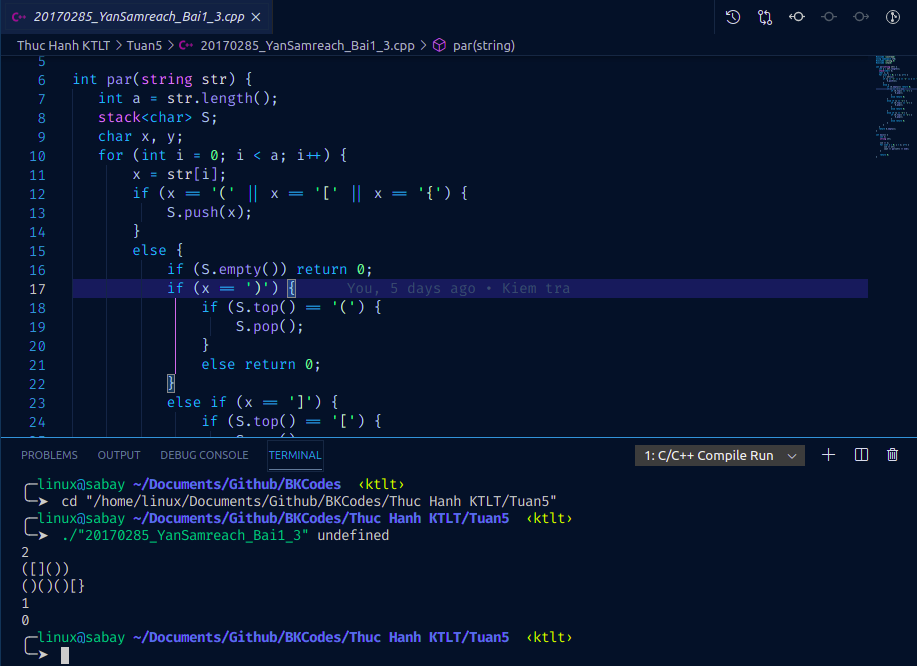
cout << par(str) << endl;

}

return 0;

}

Kết quả:



Hình 3 Bài 1.3 Dãy ngoặc đúng

### **Bài tập 4: Bài toán người du lịch**

#include <bits/stdc++.h>

#define MAX 1000

using namespace std;

int n;

int a[MAX];

int c[MAX][MAX];

bool visitted[MAX];

int c\_min = MAX\*MAX;

int f = 0;

int f\_min = MAX\*MAX;

void Try(int k){

for(int i = 2; i <= n; i++){

if(!visitted[i]){

a[k] = i;

visitted[i] = true;

f += c[a[k-1]][a[k]];

if(k == n){

f\_min = min(f + c[a[n]][a[1]], f\_min);

} else{

int g = f + (n-k+1)\*c\_min;

if(g < f\_min) Try(k+1);

}

visitted[i] = false;

f -= c[a[k-1]][a[k]];

}

}

}

int main() {

cin >> n;

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

visitted[i] = false;

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

cin >> c[i][j];

if(c[i][j] != 0) c\_min = min(c\_min, c[i][j]);

}

}

a[1] = 1;

visitted[a[1]] = true;

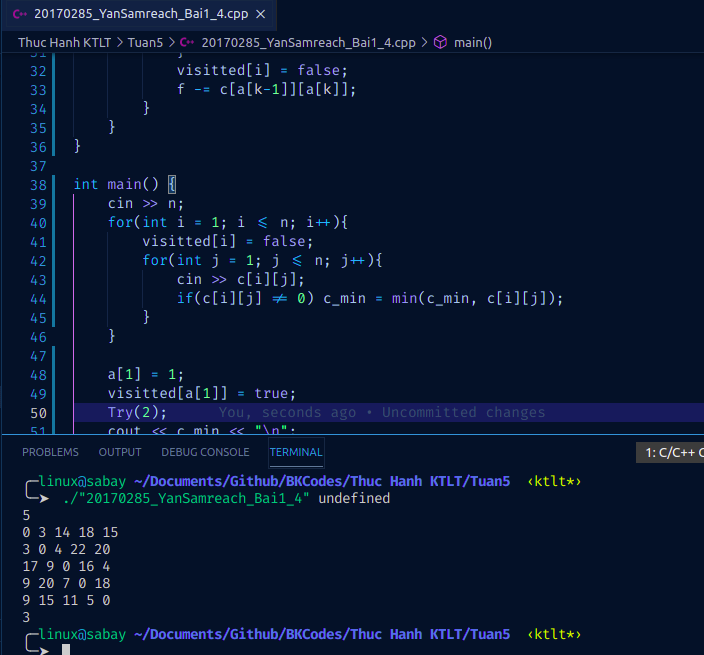
Try(2);

cout << c\_min << "\n";

return 0;

}

Kết quả:



Hình 4 Bài 1.4 Bài toán người du lịch

### **Bài tập 5: Năm nhuận**

#include <bits/stdc++.h>

using namespace std;

int main(){

bool found = false;

int a;

cin >> a;

if ((a % 4 == 0 && a % 100 != 0) || (a % 100 == 0))

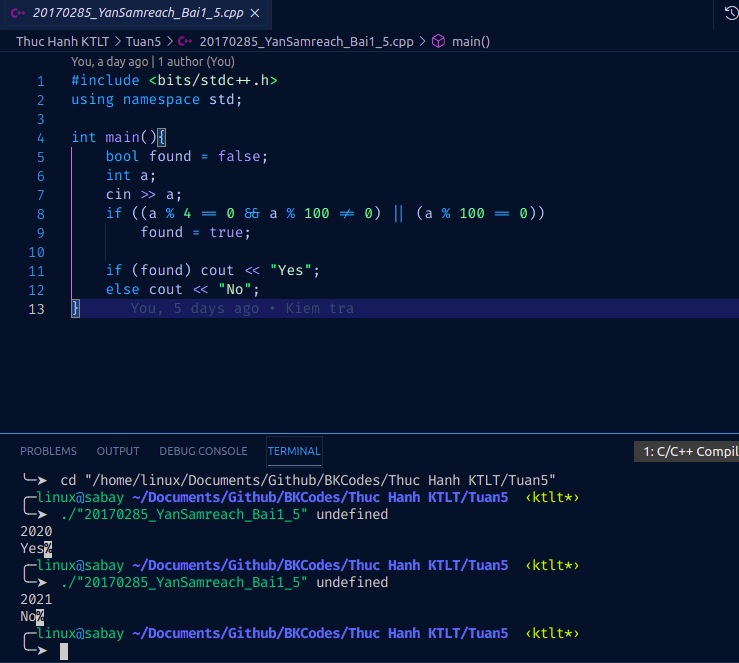
found = true;

if (found) cout << "Yes";

else cout << "No";

}

Kết quả:



Hình 5 Bài 1.5 Năm nhuận

### **Bài tập 6: Tổng kết**

Một lớp có n sinh viên. Sinh viên thứ i có điểm tổng kết là ai theo thang điểm 10. Để đánh giá chất lượng dạy học, giảng viên muốn biết có bao nhiêu bạn đạt điểm A, B, C, D, F. Quy đổi thang điểm được cho như sau:.

#include <iostream>

using namespace std;

char cal(double a) {

if (a < 4) return 'F';

if (4 <= a && a < 5.5) return 'D';

if (5.5 <= a && a < 7) return 'C';

if (7 <= a && a < 8.5) return 'B';

if (8.5 <= a) return 'A';

}

int main() {

int n;

cin >> n;

int A = 0, B = 0, C = 0, D = 0, F = 0;

while (n--) {

int a;

cin >> a;

if (cal(a) == 'A') ++A;

else if (cal(a) == 'B') ++B;

else if (cal(a) == 'C') ++C;

else if (cal(a) == 'D') ++D;

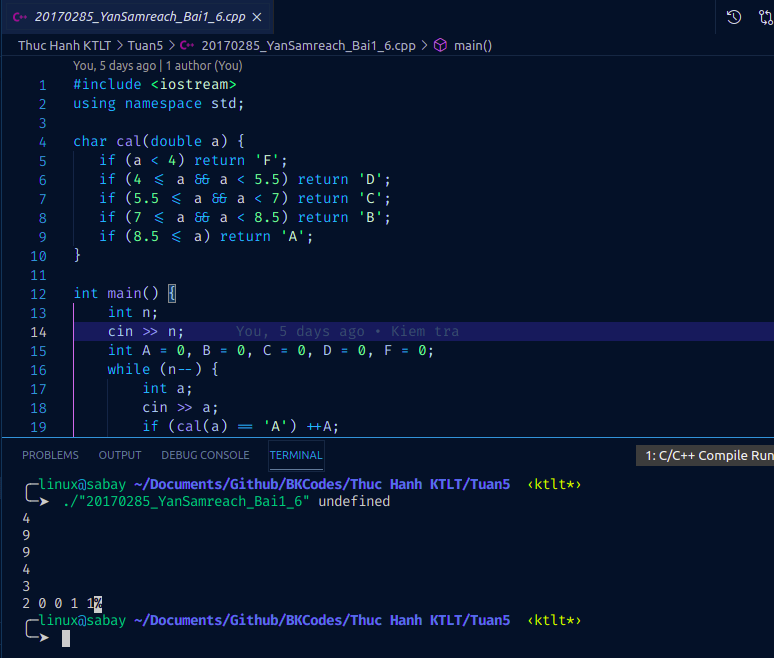
else if (cal(a) == 'F') ++F;

}

cout << A << " " << B << " " << C << " " << D << " " << F;

}

Kết quả:



Hình 6 Bài 1.6 Tổng kết

### **Bài tập 7: Chia tiền**

Sau đại dịch, thầy trò Đường Tăng muốn xin tiền của các nhà giàu để chia cho các nhà nghèo. Họ sẽ vào n thôn, thôn thứ i có ki nhà. Mỗi thôn họ sẽ quyết định xin tiền hay cho tiền, phụ thuộc vào đánh giá của họ về mức độ giàu nghèo ở đây. Nếu thôn i giàu, họ sẽ đi từng nhà trong số ki nhà này và xin ai,j tiền của nhà thứ j. Nếu thôn i nghèo, họ sẽ đi từng nhà trong số ki nhà này và phát ai,j tiền cho nhà thứ j. Hãy tính số tiền ít nhất họ phải mang theo để đảm bảo có thể phát đủ cho người nghèo (tức số tiền luôn không bị âm)

Mã nguồn sau giải quyết bài toán đó, hãy tinh chỉnh nó để tăng hiệu suất chương trình.

#include <iostream>

#include<algorithm>

using namespace std;

int main() {

int n;

cin >> n;

int ans = 0, sum = 0;

while (n--) {

int k, t;

cin >> k >> t;

while (k--) {

int a;

cin >> a;

sum += t \* a;

ans = max(ans, -sum);

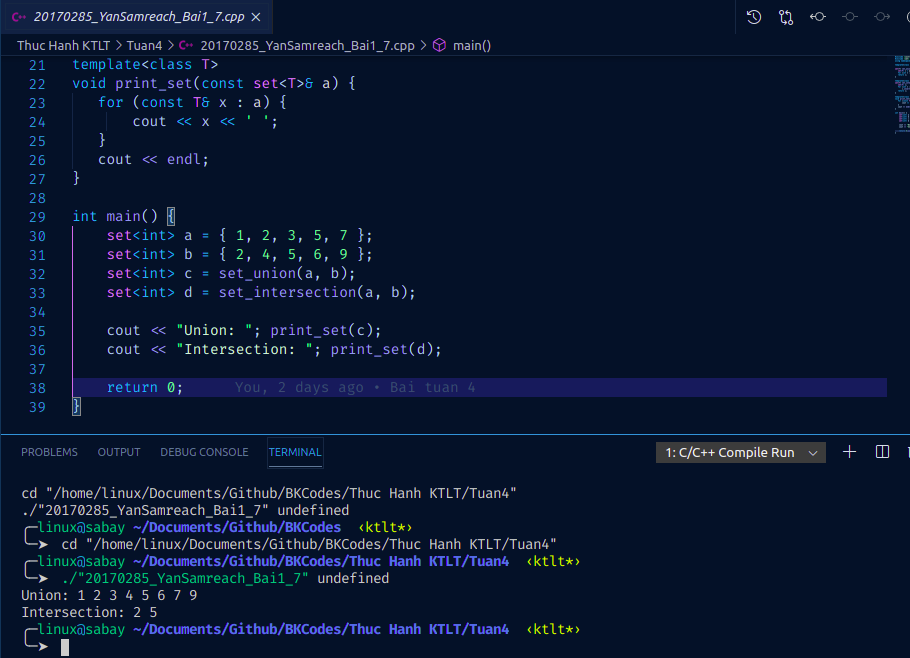
}

}

cout << ans;

}

Kết quả:



Hình 7 Bài 1.7 Chia tiền

### **Bài tập 8: Cắt hình chữ nhật**

Đề bài: <http://codeforces.com/group/Ir5CI6f3FD/contest/276073/problem/G>.

Sử dụng công cụ debug ở trên, hãy tìm và sửa các lỗi trong mã nguồn dưới đây

#include <bits/stdc++.h>

using namespace std;

struct rectangle {

int w,h;

} a[205];

int w, h, n, f[605][605];

int main() {

cin >> w >> h >> n;

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

cin >> a[i].w >> a[i].h;

}

for (int i = 1; i <= w; ++i) {

for (int j = 1; j <= h; ++j) {

f[i][j] = i\*j;

for (int k = 1; k <= n; ++k) {

if (a[k].w <= i && a[k].h <= j) {

f[i][j] = min(f[i][j], f[i][j-a[k].h] + f[i-a[k].w][a[k].h]);

f[i][j] = min(f[i][j], f[i-a[k].w][j] + f[a[k].w][j-a[k].h]);

}

}

}

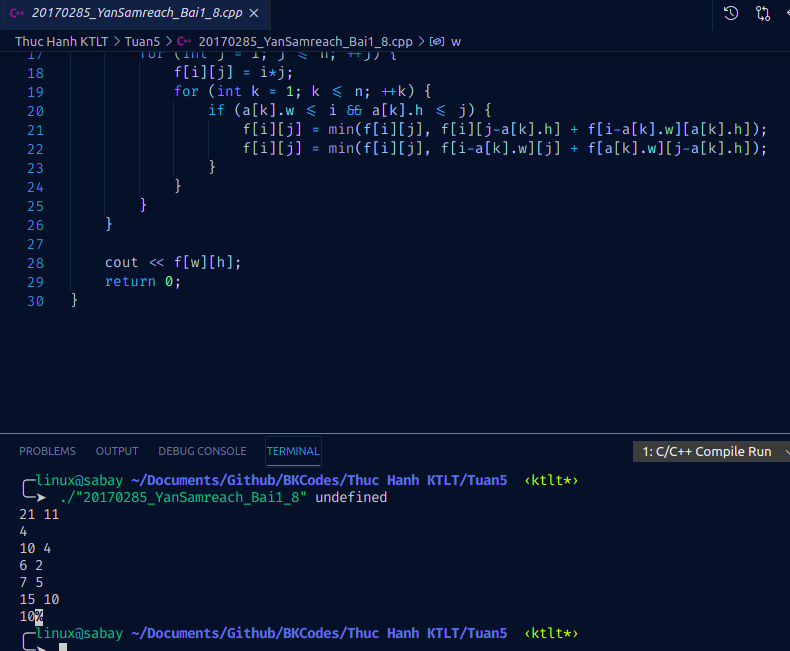
}

cout << f[w][h];

return 0;

}

Kết quả:



Hình 8 Bài 1.8 Cắt hình chữ nhật