

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 1
MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ
Giảng viên hướng dẫn: Lê Thị Hoa
MSSV: 20210909
Mã lớp: 732833

Contents

Bài thực hành số 1 – Tuần 31.....	4
Bài tập 1.1. Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trỏ tới từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.....	4
Bài tập 1.2. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: int a[7]= {13, -355, 235, 47, 67, 943, 1222};	6
Bài tập 1.3. Viết chương trình yêu cầu nhập giá trị cho 3 biến số nguyên x, y, z kiểu int. Sau đó sử dụng duy nhất một con trỏ để cộng giá trị của mỗi biến thêm 100.....	8
Bài tập 1.4. Viết hàm countEven(int*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng.	10
Bài tập 1.5. Viết hàm trả về con trỏ trỏ tới giá trị lớn nhất của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.	12
Bài tập 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. Ví dụ mảng đầu vào là [9, -1, 4, 5, 7] thì kết quả là [7, 5, 4, -1, 9].....	14
Bài tập 1.7. Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp. Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.	16
Bài tập 1.8. Viết chương trình nhập vào một ma trận 2 chiều kích thước m*n với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.....	19
Bài tập 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.....	23
Bài tập 1.10. Viết chương trình nhập vào 2 ma trận vuông cùng kích thước n*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình.....	24

Hình ảnh 1 - Bài 1.1.....	4
Hình ảnh 2 - Bài 1.1.....	5
Hình ảnh 3 - Bài 1.1.....	6
Hình ảnh 4 - Bài 1.2.....	7
Hình ảnh 5 - Bài 1.2.....	7
Hình ảnh 6 - Bài 1.3.....	26
Hình ảnh 7 - Bài 1.3.....	9
Hình ảnh 8 - Bài 1.4.....	10
Hình ảnh 9 - Bài 1.4.....	11
Hình ảnh 10 - Bài 1.5.....	12
Hình ảnh 11 - Bài 1.5.....	13
Hình ảnh 12 - Bài 1.6.....	14
Hình ảnh 13 - Bài 1.6.....	15
Hình ảnh 14 - Bài 1.6.....	15
Hình ảnh 15 - Bài 1.7.....	16
Hình ảnh 16 - Bài 1.7.....	17
Hình ảnh 17 - Bài 1.7.....	18
Hình ảnh 18 - Bài 1.7.....	18
Hình ảnh 19 - Bài 1.8.....	19
Hình ảnh 20 - Bài 1.8.....	20
Hình ảnh 21 - Bài 1.8.....	20
Hình ảnh 22 - Bài 1.8.....	21
Hình ảnh 23 - Bài 1.8.....	21
Hình ảnh 24 - Bài 1.8.....	22
Hình ảnh 25 - Bài 1.8.....	22
Hình ảnh 26 - Bài 1.9.....	23
Hình ảnh 27 - Bài 1.9.....	23
Hình ảnh 28 - Bài 1.10.....	24
Hình ảnh 29 - Bài 1.10.....	24
Hình ảnh 30 - Bài 1.10.....	25
Hình ảnh 31 - Bài 1.10.....	26

Bài thực hành số 1 – Tuần 31

Copy code (kèm ảnh code và kết quả vào)

Bài tập 1.1. Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trả về từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.

Lưu ý: Phép toán & trả về địa chỉ của biến.

Bài 1.1. Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trả về từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.

Lưu ý: Phép toán & trả về địa chỉ của biến.

For example:

Input	Result
3 4 5	Enter three integers: The three integers are: x = 3 y = 4 z = 5

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

```

1 /* 
2 Le Dinh Tri Tue - 0909 - 732833
3 Bài 1.1. Viết một chương trình C nhập vào 3 số nguyên.
4 Thiết lập một con trỏ để lần lượt trả về từng số nguyên
5 và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.
6 Lưu ý: Phép toán & trả về địa chỉ của biến.
7 */
8 # include <stdio.h>
9 int main(){
10     int x, y, z;
11     int* ptr;
12     printf("Enter three integers: ");
13     scanf("%d %d %d", &x, &y, &z);
14     printf("\nThe three integers are:\n");
15     ptr = &x;
16     printf("x = %d\n", *ptr);
17
18 //*****
19 // YOUR CODE HERE #
20 // Le Dinh Tri Tue - 0909 - 732833

```

Hình ảnh 1 - Bài 1.1

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

```
7  */
8 # include <stdio.h>
9 int main(){
10     int x, y, z;
11     int* ptr;
12     printf("Enter three integers: ");
13     scanf("%d %d %d", &x, &y, &z);
14     printf("\nThe three integers are:\n");
15     ptr = &x;
16     printf("x = %d\n", *ptr);
17
18     //*****
19     // YOUR CODE HERE #
20     // Le Dinh Tri Tue - 0909 - 732833
21     ptr = &y;
22     printf("y = %d\n", *ptr);
23     ptr = &z;
24     printf("z = %d\n", *ptr);
25     //*****
26
27     return 0;
28 }
```

Precheck

Check

Hình ảnh 2 - Bài 1.1

```

24     printf("z = %d\n", *ptr);
25     ****
26
27     return 0;
28 }
```

Precheck

Check

	Input	Expected	Got	
✓	3 4 5	Enter three integers: The three integers are: x = 3 y = 4 z = 5	Enter three integers: The three integers are: x = 3 y = 4 z = 5	✓
✓	133 24 5	Enter three integers: The three integers are: x = 133 y = 24 z = 5	Enter three integers: The three integers are: x = 133 y = 24 z = 5	✓

Passed all tests! ✓

Hình ảnh 3 - Bài 1.1

Bài tập 1.2. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: int a[7] = {13, -355, 235, 47, 67, 943, 1222};

Lưu ý:

Để in địa chỉ con trỏ các bạn sử dụng ký tự định dạng %p

Để lấy địa chỉ của một biến ta có thể dùng phép toán &

```
C Bai2.c > ⚙ main()
1  /*
2   //Lê Đình Trí Tuệ - 0909 - 732833
3   Viết chương trình in ra địa chỉ của 5 phần tử
4   đầu tiên trong mảng được định nghĩa sau đây:
5   int a[7] = {13, -355, 235, 47, 67, 943, 1222};
6  */
7
8 #include <stdio.h>
9 int main(){
10    int a[7] = {13, -355, 235, 47, 67, 943, 1222};
11    printf("address of first five elements in memory.\n");
12    for (int i=0; i<5;i++) printf("\ta[%d] ",i);
13    printf("\n");
14    //Lê Đình Trí Tuệ - 0909 - 732833
15    //*****
16    //# YOUR CODE HERE #
17    for (int i = 0; i < 5;i++) {
18        printf("%p, ", &a[i]);
19    }

```

Hình ảnh 4 - Bài 1.2

```
16 //# YOUR CODE HERE #
17 for (int i = 0; i < 5;i++) {
18     printf("%p, ", &a[i]);
19 }
20 //*****
21
22 return 0;
23 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Han
address of first five elements in memory.
a[0]    a[1]    a[2]    a[3]    a[4]
000000dbd69ff930, 000000dbd69ff934, 000000dbd69ff938, 000000dbd69ff93c, 000000dbd69ff940,
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> []
```

Hình ảnh 5 - Bài 1.2

Bài tập 1.3. Viết chương trình yêu cầu nhập giá trị cho 3 biến số nguyên x, y, z kiểu int. Sau đó sử dụng duy nhất một con trỏ để cộng giá trị của mỗi biến thêm 100.

Bài 1.3. Viết chương trình yêu cầu nhập giá trị cho 3 biến số nguyên x, y, z kiểu int. Sau đó sử dụng duy nhất một con trỏ để cộng giá trị của mỗi biến thêm 100.

```
#include <stdio.h>
int main() {
    int x, y, z;
    int *ptr;
    scanf("%d %d %d", &x, &y, &z);
    printf("Here are the values of x, y, and z:\n");
    printf("%d %d %d\n", x, y, z);

    //*****
    # YOUR CODE HERE #
    *****/
    printf("Once again, here are the values of x, y, and z:\n");
    printf("%d %d %d\n", x, y, z);
    return 0;
}
```

For example:

Input	Result
25 50 75	Here are the values of x, y, and z: 25 50 75 Once again, here are the values of x, y, and z: 125 150 175

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

```
9 | int main()
10 | {
11 |     int x, y, z;
12 |     int *ptr;
```

Hình ảnh 6 - Bài 1.3

```

12 int *ptr;
13 scanf("%d %d %d", &x, &y, &z);
14 printf("Here are the values of x, y, and z:\n");
15 printf("%d %d %d\n", x, y, z);
16
17 //*****
18 // YOUR CODE HERE #
19 ptr = &x;
20 *ptr = *ptr + 100;
21 ptr = &y;
22 *ptr = *ptr + 100;
23 ptr = &z;
24 *ptr = *ptr + 100;
25 //Le Dinh Tri Tue - 0909 - 732833|
26 //*****
27
28 printf("Once again, here are the values of x, y, and z:\n");
29 printf("%d %d %d\n", x, y, z);
30 return 0;
31 }

```

Precheck

Check

	Input	Expected	Got	
✓	25 50 75	Here are the values of x, y, and z: 25 50 75 Once again, here are the values of x, y, and z: 125 150 175	Here are the values of x, y, and z: 25 50 75 Once again, here are the values of x, y, and z: 125 150 175	✓
✓	125 150 185	Here are the values of x, y, and z: 125 150 185 Once again, here are the values of x, y, and z: 225 250 285	Here are the values of x, y, and z: 125 150 185 Once again, here are the values of x, y, and z: 225 250 285	✓

Hình ảnh 7 - Bài 1.3

Bài tập 1.4. Viết hàm countEven(int*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng.

Bài 1.4. Viết hàm countEven(int*, int) nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng???

For example:

Test	Result
int arr[] = {1, 5, 4, 8, 10, 6, 7, 2}; cout << counteven(arr, 8);	5

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

```

1 /*
2 Le Dinh Tri Tue - 0909 - 732833
3 Bài 1.4. Viết hàm countEven(int*, int) nhận một mảng số nguyên
4 và kích thước của mảng, trả về số lượng số chẵn trong mảng???
5 */
6
7 int counteven(int* arr, int size){
8
9     //*****
10    //# YOUR CODE HERE #
11    int num = 0;
12    for (int i = 0; i < size; i++){
13        if (*(arr + i) % 2 == 0){
14            num = num + 1;
15        }
16    }
17    //Le Dinh Tri Tue - 0909 - 732833
18    //*****
19
20    return num;
21 }
```

Hình ảnh 8 - Bài 1.4

```
10 //# YOUR CODE HERE #
11 int num = 0;
12 for (int i = 0; i < size; i++){
13     if (*(arr + i) % 2 == 0){
14         num = num + 1;
15     }
16 }
17 //Le Dinh Tri Tue - 0909 - 732833
18 //*****
19
20 return num;
21 }
22 
```

Precheck Check

	Test	Expected	Got	
✓	int arr[] = {1, 5, 4, 8, 10, 6, 7, 2}; cout << counteven(arr, 8);	5	5	✓
✓	int arr[] = {1, 5, 4, 0, 10, 7}; cout << counteven(arr, 6);	3	3	✓

Passed all tests! ✓

Hình ảnh 9 - Bài 1.4

Bài tập 1.5. Viết hàm trả về con trỏ trả tới giá trị lớn nhất của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.

Bài 1.5. Viết hàm trả về con trỏ trả tới giá trị lớn nhất của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.

```
double* maximum(double* a, int size){
```

```
    double *max;
```

```
    max = a;
```

```
    if (a==NULL) return NULL;
```

```
    ****
```

```
    # YOUR CODE HERE #
```

```
    ****
```

```
    return max;
```

```
}
```

For example:

Test	Result
double arr[] = {1., 10., 2., -7., 25., 3.}; double* max = maximum(arr, 6); printf("%.0f", *max);	25

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

```
1 */  
2 Le Dinh Tri Tue - 0909 - 732833  
3 Bài 1.5. Viết hàm trả về con trỏ trả tới giá trị lớn nhất  
4 của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.  
5 */  
6
```

Hình ảnh 10 - Bài 1.5

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

```

1  /*
2  Le Dinh Tri Tue - 0909 - 732833
3  Bài 1.5. Viết hàm trả về con trỏ trả về giá trị lớn nhất
4  của một mảng các số double. Nếu mảng rỗng hãy trả về NULL.
5  */
6
7  double* maximum(double* a, int size){
8      double *max;
9      max = a;
10     if (a==NULL) return NULL;
11
12     //*****#
13     //## YOUR CODE HERE ##
14     //Le Dinh Tri Tue - 0909 - 732833
15     for (int i = 0; i < size; i++){
16         if (*max < *(a + i)){
17             max = a + i;
18         }
19     }
20
21     //*****
22 }
```

Precheck

Check

	Test	Expected	Got	
✓	double arr[] = {1., 10., 2., -7., 25., 3.}; double* max = maximum(arr, 6); printf("%.0f", *max);	25	25	✓
✓	double* arr = NULL; double* max = maximum(arr, 6); printf("%.d", max==NULL?1:0);	1	1	✓
✓	double arr[] = {12.}; double* max = maximum(arr, 1); printf("%.0f", *max);	12	12	✓

Passed all tests! ✓

Hình ảnh 11 - Bài 1.5

Bài tập 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.

Ví dụ mảng đầu vào là [9, -1, 4, 5, 7] thì kết quả là [7, 5, 4, -1, 9].

Bài 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.

Ví dụ mảng đầu vào là [9, -1, 4, 5, 7] thì kết quả là [7, 5, 4, -1, 9].

```
void reversearray(int arr[], int size){
    int l = 0, r = size - 1, tmp;
    // YOUR CODE HERE #
    ****
}
```

```
void ptr_reversearray(int *arr, int size){
    int l = 0, r = size - 1, tmp;
    // YOUR CODE HERE #
    ****
}
```

For example:

Test	Result
<pre>int arr[] = {9, 3, 5, 6, 2, 5}; reversearray(arr, 6); for(int i = 0; i < 6; i++) cout << arr[i] << " "; int arr2[] = {4, -1, 5, 9}; ptr_reversearray(arr2, 4); for(int i = 0; i < 4; i++) cout << arr2[i] << " ";</pre>	5 2 6 5 3 9 9 5 -1 4

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

```
1 /*
2 Le Dinh Tri Tue - 0909 - 732833
3 Bài 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.
4 */
5
6 void reversearray(int arr[], int size){
7     //int l = 0, r = size - 1, tmp;
8     int tmp;
```

Hình ảnh 12 - Bài 1.6

```

5
6 void reversearray(int arr[], int size){
7     //int l = 0, r = size - 1, tmp;
8     int tmp;
9
10    //*****
11    //# YOUR CODE HERE #
12    //Le Dinh Tri Tue - 0909 - 732833
13
14    //*****
15    for (int i = 0; i < size; i++){
16        for (int j = i + 1; j < size; j++){
17            tmp = arr[i];
18            arr[i] = arr[j];
19            arr[j] = tmp;
20        }
21    }
22 }
23 }
```

Hình ảnh 13 - Bài 1.6

```

24
25 void ptr_reversearray(int *arr, int size){
26     //int l = 0, r = size - 1, tmp;
27     int tmp;
28
29     //*****
30     //# YOUR CODE HERE #
31     //Le Dinh Tri Tue - 0909 - 732833
32     for (int i = 0; i < size; i++){
33         for (int j = i + 1; j < size; j++){
34             tmp = *(arr + i);
35             *(arr + i) = *(arr + j);
36             *(arr + j) = tmp;
37         }
38     }
39 }
40 //*****
41 }
```

[Precheck](#) [Check](#)

	Test	Expected	Got	
✓	<pre> int arr[] = {9, 3, 5, 6, 2, 5}; reversearray(arr, 6); for(int i = 0; i < 6; i++) cout << arr[i] << " "; int arr2[] = {4, -1, 5, 9}; ptr_reversearray(arr2, 4); for(int i = 0; i < 4; i++) cout << arr2[i] << " "; </pre>	5 2 6 5 3 9 9 5 -1 4	5 2 6 5 3 9 9 5 -1 4	✓
✓	<pre> int arr[] = {9, 6, 2, 5}; reversearray(arr, 4); for(int i = 0; i < 4; i++) cout << arr[i] << " "; int arr2[] = {12, 4, -1, 5, 9}; ptr_reversearray(arr2, 5); for(int i = 0; i < 5; i++) cout << arr2[i] << " "; </pre>	5 2 6 9 9 5 -1 4 12	5 2 6 9 9 5 -1 4 12	✓

Passed all tests! ✓

Hình ảnh 14 - Bài 1.6

Bài tập 1.7. Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp.

Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.

Bài 1.7. Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp.

Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.

For example:

Input	Result
5 9 -2 1 2 15	Enter the number of elements: The input array is: 9 -2 1 2 15 The sorted array is: -2 1 2 9 15

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

```

1 /*  

2 Le Dinh Tri Tue - 0909 - 732833  

3 Bai 1.7. Viết chương trình nhập vào một mảng các số nguyên với số lượng  

4 các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần.  

5 Hiển thị danh sách mảng trước và sau khi sắp xếp.  

6 Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.  

7 */  

8 |  

9 #include <stdio.h>  

10 #include <cstdlib>  

11 int *a;  

12 int n, tmp;  

13  

14 int main(){  

15     printf("Enter the number of elements: ");  

16     scanf("%d", &n);  

17  

18     //Allocate memory  

19     //Le Dinh Tri Tue - 0909 - 732833  

20     //*****  

21     //# YOUR CODE HERE #  

22     a = (int*)calloc(n, sizeof(int));
```

Hình ảnh 15 - Bài 1.7

```
22 // " YOUR CODE HERE "
23 a = (int*)calloc(n, sizeof(int));
24 //*****
25
26 for(int i = 0; i < n; i++)
27     scanf("%d", a + i);
28
29 printf("The input array is: \n");
30 for(int i = 0; i < n; i++)
31     printf("%d ", *(a + i));
32 printf("\n");
33
34 //Sort array
35 //Le Dinh Tri Tue - 0909 - 732833
36
37 //*****
38 //# YOUR CODE HERE #
39 for (int i = 0; i < n; i++){
40     for (int j = i + 1; j < n; j++){
41         if (*(a + i) > *(a + j)){
42             tmp = *(a + i);
```

Hình ảnh 16 - Bài 1.7

```

40   }
41   for (int j = i + 1; j < n; j++) {
42       if (*(a + i) > *(a + j)){
43           tmp = *(a + i);
44           *(a + i) = *(a + j);
45           *(a + j) = tmp;
46       }
47   }
48 }
49 //*****
50
51 printf("The sorted array is: \n");
52 for(int i = 0; i < n; i++)
53     printf("%d ", *(a + i));
54 printf("\n");
55 //Le Dinh Tri Tue - 0909 - 732833
56
57 delete [] a;
58 return 0;
59 }
```

Hình ảnh 17 - Bài 1.7

```

48 }
49 //*****
50
51 printf("The sorted array is: \n");
52 for(int i = 0; i < n; i++)
53     printf("%d ", *(a + i));
54 printf("\n");
55 //Le Dinh Tri Tue - 0909 - 732833
56
57 delete [] a;
58 return 0;
59 }
```

[Precheck](#) [Check](#)

	Input	Expected	Got
✓	5 9 -2 1 2 15	Enter the number of elements: The input array is: 9 -2 1 2 15 The sorted array is: -2 1 2 9 15	Enter the number of elements: The input array is: 9 -2 1 2 15 The sorted array is: -2 1 2 9 15
✓	10 9 -2 1 2 15 3 7 8 20 -5	Enter the number of elements: The input array is: 9 -2 1 2 15 3 7 8 20 -5 The sorted array is: -5 -2 1 2 3 7 8 9 15 20	Enter the number of elements: The input array is: 9 -2 1 2 15 3 7 8 20 -5 The sorted array is: -5 -2 1 2 3 7 8 9 15 20

Passed all tests! ✓

Hình ảnh 18 - Bài 1.7

Bài tập 1.8. Viết chương trình nhập vào một ma trận 2 chiều kích thước $m \times n$ với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.

Lưu ý: Khi viết hàm cấp phát bộ nhớ cho một ma trận hai chiều biểu diễn bởi con trỏ `int **mt`, nếu ta truyền con trỏ theo kiểu địa chỉ `void allocate(int **mt, int m, int n)` sẽ dẫn tới việc cấp phát bộ nhớ cho một bản sao của con trỏ `**mt`. Do đó, sau khi gọi hàm thì con trỏ `**mt` gốc vẫn không được cấp phát bộ nhớ. Để cấp phát thành công cần truyền con trỏ theo dạng địa chỉ, ví dụ sử dụng con trỏ cấp 3 dạng `int ***mt`.

Bài 1.8. Viết chương trình nhập vào một ma trận 2 chiều kích thước $m \times n$ với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.

Lưu ý: Khi viết hàm cấp phát bộ nhớ cho một ma trận hai chiều biểu diễn bởi con trỏ `int ***mt`, nếu ta truyền con trỏ theo kiểu địa chỉ `void allocate_mem(int **mt, int m, int n)` sẽ dẫn tới việc cấp phát bộ nhớ cho một bản sao của con trỏ `**mt`. Do đó, sau khi gọi hàm thì con trỏ `**mt` gốc vẫn không được cấp phát bộ nhớ. Để cấp phát thành công cần truyền con trỏ cấp 3 dạng `int ***mt`.

For example:

Input	Result
2 2	Enter m, n = mt[0][0] = mt[0][1] = mt[1][0] = mt[1][1] = 1 2
1 2	4 5
4 5	The sum of all even elements is 6

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

```

1 /*
2 Le Dinh Tri Tue - 0909 - 732833
3 Bài 1.8. Viết chương trình nhập vào một ma trận 2 chiều kích thước m*n
4 với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.
5 Lưu ý: Khi viết hàm cấp phát bộ nhớ cho một ma trận hai chiều biểu diễn bởi
6 con trỏ int ***mt, nếu ta truyền con trỏ theo kiểu địa chỉ
7 void allocate_mem(int ***mt, int m, int n) sẽ dẫn tới việc cấp phát bộ nhớ cho một
8 bản sao của con trỏ **mt. Do đó, sau khi gọi hàm thì con trỏ **mt gốc vẫn không
9 được cấp phát bộ nhớ. Để cấp phát thành công cần truyền con trỏ theo dạng địa chỉ,
10 ví dụ sử dụng con trỏ cấp 3 dạng int ***mt.
11 */
12 #include <stdio.h>
13 #include <cstdlib>
14
15 void allocate_mem(int ***mt, int m, int n){
16
17     //Allocate memory for the matrix
18
19     //*****
20     //## YOUR CODE HERE #
21     //Le Dinh Tri Tue - 0909 - 732833
22     *mt = (int**)malloc(m * sizeof(int *));

```

Hình ảnh 19 - Bài 1.8

```

14
15 void allocate_mem(int ***mt, int m, int n){
16
17     //Allocate memory for the matrix
18
19     //*****
20     //# YOUR CODE HERE #
21     //Le Dinh Tri Tue - 0909 - 732833
22     *mt = (int**)malloc(m * sizeof(int *));
23     for (int i = 0; i < m; i++){
24         (*mt)[i] = (int*)malloc(n * sizeof(int));
25     }
26     //*****
27 }
28
29
30 void input(int **mt, int m, int n){
31     //Input elements of the matrix
32
33     //*****
34     //# YOUR CODE HERE #
35     //Le Dinh Tri Tue - 0909 - 732833
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

Hình ảnh 20 - Bài 1.8

```

28
29
30 void input(int **mt, int m, int n){
31     //Input elements of the matrix
32
33     //*****
34     //# YOUR CODE HERE #
35     //Le Dinh Tri Tue - 0909 - 732833
36     for (int i = 0; i < m; i++){
37         for (int j = 0; j < n; j++){
38             int tempp;
39             scanf("%d", &tempp);
40             *(mt[i] + j) = tempp;
41         }
42     }
43     //*****
44 }
45
46 void output(int **mt, int m, int n){
47     // Print all elements of the matrix
48
49

```

Hình ảnh 21 - Bài 1.8

```

44 |
45 |
46 void output(int **mt, int m, int n){
47     ## Print all elements of the matrix
48
49     //*****
50     ## YOUR CODE HERE #
51     //Le Dinh Tri Tue - 0909 - 732833
52     for (int i = 0; i < m; i++){
53         for (int j = 0; j < n; j++){
54             printf("mt[%d][%d] = ", i, j);
55         }
56     }
57     for (int i = 0; i < m; i++){
58         for (int j = 0; j < n; j++){
59             printf("%d ", *(mt[i] + j));
60         }
61         printf("\n");
62     }
63
64     //*****
65 }
-- 

```

Hình ảnh 22 - Bài 1.8

```

-- 
67 |
68 int process(int **mt, int m, int n){
69     int tong = 0;
70     ## Calculate the sum of all even elements in the matrix
71
72     //*****
73     ## YOUR CODE HERE #
74     //Le Dinh Tri Tue - 0909 - 732833
75     for (int i = 0; i < m; i++){
76         for (int j = 0; j < n; j++){
77             if (*(mt[i] + j) % 2 == 0){
78                 tong = tong + *(mt[i] + j);
79             }
80         }
81     }
82     //*****
83
84     return tong;
85 }
-- 

```

Hình ảnh 23 - Bài 1.8

```

85 }
86
87 void free_mem(int **mt, int m, int n){
88     /* Free memory
89
90     *****
91     /* YOUR CODE HERE #
92     //Le Dinh Tri Tue - 0909 - 732833
93     for (int i = 0; i < m; i++){
94         free((mt)[i]);
95     }
96     free(mt);
97     *****/
98 }
99
100 int main(){
101     int m, n, **mt;
102     printf("Enter m, n = ");

```

Hình ảnh 24 - Bài 1.8

```

98 }   ''
99
100 int main(){
101     int m, n, **mt;
102     printf("Enter m, n = ");
103     scanf("%d%d", &m, &n);
104     allocate_mem(&mt, m, n);
105     input(mt, m, n);
106     output(mt, m, n);
107     printf("The sum of all even elements is %d", process(mt, m, n));
108     free_mem(mt, m, n);
109     return 0;
110 }

```

Precheck**Check**

	Input	Expected	Got
✓	2 2 1 2 4 5	Enter m, n = mt[0][0] = mt[0][1] = mt[1][0] = mt[1][1] = 1 2 4 5 The sum of all even elements is 6	Enter m, n = mt[0][0] = mt[0][1] = mt[1][0] = mt[1][1] = 4 5 The sum of all even elements is 6

Passed all tests! ✓

Hình ảnh 25 - Bài 1.8

Bài tập 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.

```

/*
Lê Đình Trí Tuệ - 0909 - 732833
Bai 9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước
*/
#include <stdio.h>

void printAll(int *array, int n)
{
    for (int i = 0; i < n; i++){
        for (int j = i + 1; j < n; j++){
            for (int k = i; k < j; k++){
                printf("%d ", *(array + k));
            }
            printf("\n");
        }
    }
}

//Lê Đình Trí Tuệ - 0909 - 732833
int main(){
    int array[] = {1, 3, 4, 2};
    printAll(array, 5);
    return 0;
}

```

Hình ảnh 26 - Bài 1.9

```

17 }
18 //Lê Đình Trí Tuệ - 0909 - 732833
19 int main(){
20     int array[] = {1, 3, 4, 2};
21     printAll(array, 5);
22     return 0;
23 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\"
}
1
1 3
1 3 4
1 3 4 2
3
3 4
3 4 2
4
4 2
2
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh>

```

Hình ảnh 27 - Bài 1.9

Bài tập 1.10. Viết chương trình nhập vào 2 ma trận vuông cùng kích thước $n \times n$, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình.

Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận.

```
/*
Lê Đình Trí Tuệ - 0909 - 732833
Bai 10. Viết chương trình nhập vào 2 ma trận vuông
cùng kích thước  $n \times n$ , trong đó  $n$  nhập từ bàn phím.
Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình.
Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận.
*/

#include <stdio.h>
#include <stdlib.h>
void sum2Arr(int **arr1, int **arr2, int n){
    for (int i = 0; i < n; i++){
        for (int j = 0; j < n; j++){
            //int temp = arr1[i][j] + arr2[i][j];
            int temp = *(arr1[i] + j) + *(arr2[i] + j);
            printf("%d ", temp);
        }
        printf("\n");
    }
}
//Lê Đình Trí Tuệ - 0909 - 732833
```

Hình ảnh 28 - Bài 1.10

```
1 //Lê Đình Trí Tuệ - 0909 - 732833
2 void multi2Arr(int **arr1, int **arr2, int n){
3     for (int i = 0; i < n; i++){
4         for (int j = 0; j < n; j++){
5             int temp = 0;
6             for (int k = 0; k < n; k++){
7                 //temp = temp +
8                 temp = temp + (*(arr1[i] + k)) * (*(arr2[k] + j));
9             }
10            printf("%d ", temp);
11        }
12        printf("\n");
13    }
14 }
//Lê Đình Trí Tuệ - 0909 - 732833
15 int main(){
16     int n;
17     scanf("%d", &n);
18     int **array1 = (int **)calloc(n, sizeof(int *));
19     for (int i = 0; i < n; i++) {
20         array1[i] = (int *)calloc(n, sizeof(int));
21     }
22 }
```

Hình ảnh 29 - Bài 1.10

```
13     int **array2 = (int **)calloc(n, sizeof(int *));
14     for (int i = 0; i < n; i++) {
15         array2[i] = (int *)calloc(n, sizeof(int));
16     }
17 //Lê Đình Trí Tuệ - 0909 - 732833
18     for (int i = 0; i < n; i++){
19         for (int j = 0; j < n; j++){
20             int temp;
21             scanf("%d", &temp);
22             array1[i][j] = temp;
23         }
24     }
25 //Lê Đình Trí Tuệ - 0909 - 732833
26     for (int i = 0; i < n; i++){
27         for (int j = 0; j < n; j++){
28             int temp;
29             scanf("%d", &temp);
30             array2[i][j] = temp;
31         }
32     }
33     sum2Arr(array1, array2, n);
34     multi2Arr(array1, array2, n);
35     return 0;
36 }
```

Hình ảnh 30 - Bài 1.10

```
54      f
55      //Lê Đình Trí Tuệ - 0909 - 732833
56      for (int i = 0; i < n; i++){
57          for (int j = 0; j < n; j++){
58              int temp;
59              scanf("%d", &temp);
60              array2[i][j] = temp;
61      }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
i10 }
4
1 2 3 5
2 6 5 3
0 -5 9 7
-3 -5 7 -1
2 5 8 9
4 6 5 8
1 1 1 1
2 2 5 4
3 7 11 14
6 12 10 11
1 -4 10 8
-1 -3 12 3
23 30 46 48
39 57 66 83
3 -7 19 -3
-21 -40 -47 -64
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh>
```

Hình ảnh 31 - Bài 1.10

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 2
MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ

MSSV: 20210909

Mã lớp: 732833

Contents

Bài thực hành số 2 – Tuần 11.....	30
Bài tập 2.1. Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.....	30
Bài tập 2.2. Viết hàm hoán vị vòng tròn 3 biến a, b, c. Sau khi thực hiện hàm, các biến a, b, c tương ứng nhận các giá trị mới b, c, a.....	31
Bài tập 2.3. Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị ax^2+bx+c với a, b, c định sẵn.....	33
Bài tập 2.4. Viết các hàm tính lập phương của số nguyên và số thực.....	35
Bài tập 2.5. Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức.....	37
Bài tập 2.6. Giả thuyết Collatz: bắt đầu từ số dương n bất kỳ, nếu n chẵn thì chia 2, nếu lẻ thì nhân 3 cộng 1, giả thuyết cho rằng ta luôn đi đến n=1. Hãy viết chương trình mô phỏng lại quá trình biến đổi để kiểm chứng giả thuyết với giá trị của n nhập từ bàn phím.....	40
Bài tập 2.7. Viết hàm tính tổng các phần tử trong hai mảng. Yêu cầu sử dụng function template để cho phép hàm làm việc với các mảng số nguyên lẫn số thực.....	44
Bài tập 2.8. Viết hàm so sánh cho thuật toán sắp xếp.....	46
Bài tập 2.9. Dưới đây cung cấp đoạn code đơn giản để tính hàm sigmoid theo công thức trực tiếp. Hãy viết hàm tính xấp xỉ sigmoid(x) đến độ chính xác 10–6 và có tốc độ nhanh hơn ít nhất 30% so với code đơn giản.....	48
Bài tập 2.10 (BONUS). Dưới đây cung cấp đoạn code đơn giản để tính tích của hai ma trận cỡ NxN theo công thức trực tiếp. Hãy viết hàm tính tích hai ma trận nhưng có tốc độ nhanh hơn ít nhất 10% so với code đơn giản. Gợi ý: hãy để ý đến thứ tự truy cập các phần tử trong ma trận, tối ưu cache hoặc sử dụng thuật toán tốt hơn O(N ³)	51
Bài tập 2.11. Cho 2 đa thức A(x) và B(x) tương ứng có bậc N và M . Hãy tính ma trận tích C(x) = A(x) * B(x) có bậc N+M–1 . Input: Gồm 2 dòng biểu diễn các đa thức A(x) và B(x), mỗi dòng Số đầu tiên N là bậc của đa thức; N+1 số nguyên tiếp theo, số thứ i là hệ số của xi–1 . Output: Một số nguyên duy nhất là XOR của các hệ số của đa thức C(x).....	60
Bài tập 2.12. Hôm nay, cô giáo giao cho An một câu hỏi hóc búa. Cô cho một danh sách với mỗi phần tử có dạng <code><key, value></code> và yêu cầu An sắp xếp danh sách đó giảm dần theo giá trị value. Nếu 2 phần tử có value giống nhau thì sắp xếp giảm dần theo key. Hãy viết một chương trình sử dụng hàm nặc danh để giúp An làm bài tập. Input: Danh sách đầu vào. Mỗi dòng ghi một cặp giá trị key, value cách nhau bởi dấu cách ($ key \leq 109$, $ value \leq 109$). Output: In danh sách đã được sắp xếp theo yêu cầu. Mỗi dòng ghi một cặp giá trị key, value cách nhau bởi dấu cách.....	62
Bài tập 2.13. Số nguyên lớn là các số nguyên có giá trị rất lớn và không thể biểu diễn bằng các kiểu dữ liệu nguyên cơ bản. Để biểu diễn số nguyên lớn, ta có thể dùng kiểu struct như sau: struct bigNum{ char sign; char num[101]; }; Nhiệm vụ các bạn là đa năng hóa các toán tử để thực hiện các phép toán số học với kiểu dữ liệu số nguyên lớn vừa định nghĩa ở trên. Input: Dữ liệu vào gồm hai dòng mô tả hai số nguyên lớn a và b , mỗi dòng chứa 1 chuỗi ký tự mô tả 1 số nguyên lớn không vượt quá 10100 . Chữ số đầu của mỗi chuỗi ký tự sẽ thể hiện dấu của số đó: 0 là âm, 1 là dương. Các chữ số sau thể hiện giá trị của số đó. Output: In ra giá trị của biểu thức $ab - 3a + 4b$. Kết quả in ra một số nguyên lớn dưới dạng chuỗi ký tự có định dạng như mô tả trong dữ liệu vào.....	67

Figure 1	30
Figure 2	30
Figure 3	31
Figure 4	32
Figure 5	33
Figure 6	34
Figure 7	35
Figure 8	36
Figure 9	37
Figure 10	37
Figure 11	38
Figure 12	38
Figure 13	39
Figure 14	40
Figure 15	41
Figure 16	42
Figure 17	43
Figure 18	44
Figure 19	44
Figure 20	45
Figure 21	46
Figure 22	47
Figure 23	48
Figure 24	48
Figure 25	49
Figure 26	49
Figure 27	50
Figure 28	51
Figure 29	61
Figure 30	62
Figure 31	64
Figure 32	65
Figure 33	66
Figure 34	67
Figure 35	76

Bài thực hành số 2 – Tuần 12

Copy code (kèm ảnh code và kết quả vào)

Bài tập 2.1. Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.

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

```

1 * /*
2 Bài 2.1. Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.
3 //Le Dinh Tri Tue - 0909 - 732833
4 */
5
6 #include <stdio.h>
7 #include <math.h>
8 //Le Dinh Tri Tue - 0909 - 732833
9 // Hàm tính độ dài cạnh huyền
10 float get_hypotenuse(float x, float y) {
11     //*****
12     //# YOUR CODE HERE #
13     return sqrt(pow(x, 2) + pow(y, 2));
14     //*****
15 }
16 //Le Dinh Tri Tue - 0909 - 732833
17 int main(){
18     float x, y;
19     scanf("%f%f", &x, &y);
20
21     float z = get_hypotenuse(x, y);
22     printf("z = %.2f\n", z);

```

Figure 32

```

12     //# YOUR CODE HERE #
13     return sqrt(pow(x, 2) + pow(y, 2));
14     //*****
15 }
16 //Le Dinh Tri Tue - 0909 - 732833
17 int main(){
18     float x, y;
19     scanf("%f%f", &x, &y);
20
21     float z = get_hypotenuse(x, y);
22     printf("z = %.2f\n", z);
23
24     return 0;
25 }

```

Precheck Check

	Input	Expected	Got	
✓	3 4	z = 5.00	z = 5.00	✓
✓	5 6	z = 7.81	z = 7.81	✓

Passed all tests! ✓

Figure 33

Bài tập 2.2. Viết hàm hoán vị vòng tròn 3 biến a, b, c. Sau khi thực hiện hàm, các biến a, b, c tương ứng nhận các giá trị mới b, c, a.

```
1 /*  
2 Viết hàm hoán vị vòng tròn 3 biến a, b, c.  
3 Sau khi thực hiện hàm, các biến a, b, c  
4 tương ứng nhận các giá trị mới b, c, a.  
5 //Le Dinh Tri Tue - 0909 - 732833  
6 */  
7  
8 #include <stdio.h>  
9 //Le Dinh Tri Tue - 0909 - 732833  
10 // Hàm rotate  
11 void rotate(int &x, int &y, int &z) {  
12     //*****  
13     # YOUR CODE HERE #  
14     //*****  
15     int temp = x;  
16     x = y; y = z; z = temp;  
17 }  
18 //Le Dinh Tri Tue - 0909 - 732833  
19 int main() {  
20     int x, y, z;  
21  
22     //# Nhập 3 số nguyên
```

Figure 34

```
16     x = y; y = z; z = temp;
17 }
18 //Le Dinh Tri Tue - 0909 - 732833
19 int main() {
20     int x, y, z;
21
22     //# Nhập 3 số nguyên
23 ****
24 # YOUR CODE HERE #
25 ****
26 scanf("%d%d%d", &x, &y, &z);
27 //Le Dinh Tri Tue - 0909 - 732833
28 printf("Before: %d, %d, %d\n", x, y, z);
29 rotate(x, y, z);
30 printf("After: %d, %d, %d\n", x, y, z);
31
32     return 0;
33 }
```

Precheck

Check

	Input	Expected	Got	
✓	3 4 5	Before: 3, 4, 5 After: 4, 5, 3	Before: 3, 4, 5 After: 4, 5, 3	✓
✓	5 7 9	Before: 5, 7, 9 After: 7, 9, 5	Before: 5, 7, 9 After: 7, 9, 5	✓

Passed all tests! ✓

Figure 35

Bài tập 2.3. Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị ax^2+bx+c với a, b, c định sẵn.

```
1 /*  
2 Bài 2.3. Viết chương trình yêu cầu nhập giá trị cho số nguyên x  
3 nhỏ hơn 100. In ra giá trị  $ax^2+bx+c$  với a, b, c định sẵn  
4 //Le Dinh Tri Tue - 0909 - 732833  
5 */  
6  
7 #include <stdio.h>  
8  
9 //Le Dinh Tri Tue - 0909 - 732833  
10 //## Viết hàm get_value  
11 ****  
12 # YOUR CODE HERE #  
13 *****/  
14  
15 int get_value(int x, int a = 2, int b = 1, int c = 0){  
16     return a*x*x + b*x + c;  
17 }  
18 //Le Dinh Tri Tue - 0909 - 732833  
19 int main(){  
20     int x;  
21     scanf("%d", &x);  
22 }
```

Figure 36

```

17 }
18 //Le Dinh Tri Tue - 0909 - 732833
19 int main(){
20     int x;
21     scanf("%d", &x);
22
23     int a = 2; //# giá trị mặc định của a
24     int b = 1; //# giá trị mặc định của b
25     int c = 0; //# giá trị mặc định của c
26     //# Nhập 3 số nguyên a, b, c từ bàn phím
27 ****
28 # YOUR CODE HERE #
29 ****
30 scanf("%d%d%d", &a, &b, &c);
31 //Le Dinh Tri Tue - 0909 - 732833
32 printf("a=2, b=1, c=0: %d\n", get_value(x));
33 printf("a=%d, b=1, c=0: %d\n", a, get_value(x, a));
34 printf("a=%d, b=%d, c=0: %d\n", a, b, get_value(x, a, b));
35 printf("a=%d, b=%d, c=%d: %d\n", a, b, c, get_value(x, a, b, c));
36
37 return 0;
38 }
```

Precheck

Check

	Input	Expected	Got	
✓	5 3 7 8	a=2, b=1, c=0: 55 a=3, b=1, c=0: 80 a=3, b=7, c=0: 110 a=3, b=7, c=8: 118	a=2, b=1, c=0: 55 a=3, b=1, c=0: 80 a=3, b=7, c=0: 110 a=3, b=7, c=8: 118	✓
✓	9 -1 5 -3	a=2, b=1, c=0: 171 a=-1, b=1, c=0: -72 a=-1, b=5, c=0: -36 a=-1, b=5, c=-3: -39	a=2, b=1, c=0: 171 a=-1, b=1, c=0: -72 a=-1, b=5, c=0: -36 a=-1, b=5, c=-3: -39	✓

Passed all tests! ✓

Figure 37

Bài tập 2.4. Viết các hàm tính lập phương của số nguyên và số thực.

```
1 /*  
2 Bài 2.4. Viết các hàm tính lập phương của số nguyên và số thực.  
3 Le Dinh Tri Tue - 0909 - 732833  
4 */  
5  
6 #include <stdio.h>  
7 #include <math.h>  
8 //Le Dinh Tri Tue - 0909 - 732833  
9 int cube(int x) {  
10     //# trả về lập phương của x  
11     /*****  
12     # YOUR CODE HERE #  
13     *****/  
14     return pow(x, 3);  
15 }  
16 double cube(double x){  
17     return pow(x, 3);  
18 }  
19 //Le Dinh Tri Tue - 0909 - 732833  
20 //# viết hàm tính lập phương của một số kiểu double  
21 /*****  
22 # YOUR CODE HERE #
```

Figure 38

```
18 }
19 //Le Dinh Tri Tue - 0909 - 732833
20 //# viết hàm tính lập phương của một số kiểu double
21 ****
22 # YOUR CODE HERE #
23 ****
24
25 int main() {
26     int n;
27     double f;
28     scanf("%d %lf", &n, &f);
29     //Le Dinh Tri Tue - 0909 - 732833
30     printf("Int: %d\n", cube(n));
31     printf("Double: %.2lf\n", cube(f));
32
33     return 0;
34 }
```

PrecheckCheck

	Input	Expected	Got	
✓	3 5.2	Int: 27 Double: 140.61	Int: 27 Double: 140.61	✓
✓	10 7.12	Int: 1000 Double: 360.94	Int: 1000 Double: 360.94	✓

Passed all tests! ✓

Figure 39

Bài tập 2.5. Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức.

```

1 /*Bài 2.5. Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức
2 //Le Dinh Tri Tue - 0909 - 732833
3 */
4 #include <iostream>
5 #include <ostream>
6 #include <math.h>
7 #include <iomanip>
8
9 using namespace std;
10 //Le Dinh Tri Tue - 0909 - 732833
11 // Struct của 1 số phức
12 struct Complex {
13     double real;
14     double imag;
15 };
16 //Le Dinh Tri Tue - 0909 - 732833
17 //Triển khai operator +
18 Complex operator + (Complex a, Complex b) {
19     *****
20     # YOUR CODE HERE #
21     *****/
22 }
```

Figure 40

```

17 //Triển khai operator +
18 Complex operator + (Complex a, Complex b) {
19     *****
20     # YOUR CODE HERE #
21     *****/
22     Complex x;
23     x.real = a.real + b.real;
24     x.imag = a.imag + b.imag;
25     return x;
26 }
27 //Le Dinh Tri Tue - 0909 - 732833
28 //Triển khai operator -
29 Complex operator - (Complex a, Complex b) {
30     *****
31     # YOUR CODE HERE #
32     *****/
33     Complex x;
34     x.real = a.real - b.real;
35     x.imag = a.imag - b.imag;
36     return x;
37 }
```

Figure 41

```

58 //Le Dinh Tri Tue - 0909 - 732833
39 // Triển khai operator *
40 v Complex operator * (Complex a, Complex b) {
41 v     ****
42     # YOUR CODE HERE #
43     ****
44     Complex x;
45     x.real = a.real*b.real - a.imag*b.imag;
46     x.imag = a.real*b.imag + a.imag*b.real;
47     return x;
48 }
49 //Le Dinh Tri Tue - 0909 - 732833
50 // Triển khai operator /
51 v Complex operator / (Complex a, Complex b) {
52 v     ****
53     # YOUR CODE HERE #
54     ****
55     Complex x;
56     x.real = (a.real*b.real + a.imag*b.imag)/(pow(b.real, 2) + pow(b.imag, 2));
57     x.imag = (-a.real*b.imag + a.imag*b.real)/(pow(b.real, 2) + pow(b.imag, 2));
58     return x;
59 }

```

Figure 42

```

59 }
60 //Le Dinh Tri Tue - 0909 - 732833
61 v ostream& operator << (ostream& out, const Complex &a) {
62     out << '(' << std::setprecision(2) << a.real << (a.imag >= 0 ? '+' : '-')
63     return out;
64 }
65 //Le Dinh Tri Tue - 0909 - 732833
66 v int main() {
67     double real_a, real_b, img_a, img_b;
68     // Nhập vào phần thực và phần ảo của 2 số phức a và bb
69     cin >> real_a >> img_a;
70     cin >> real_b >> img_b;
71     //Le Dinh Tri Tue - 0909 - 732833
72     Complex a{real_a, img_a};
73     Complex b{real_b, img_b};
74     //Le Dinh Tri Tue - 0909 - 732833
75     cout << a << " + " << b << " = " << a + b << endl;
76     cout << a << " - " << b << " = " << a - b << endl;
77     cout << a << " * " << b << " = " << a * b << endl;
78     cout << a << " / " << b << " = " << a / b << endl;
79
80

```

Figure 43

```

73     Complex b{real_b, img_b};
74     //Le Dinh Tri Tue - 0909 - 732833
75     cout << a << " + " << b << " = " << a + b << endl;
76     cout << a << " - " << b << " = " << a - b << endl;
77     cout << a << " * " << b << " = " << a * b << endl;
78     cout << a << " / " << b << " = " << a / b << endl;
79
80     return 0;
81 }
```

Precheck**Check**

	Input	Expected	Got	
✓	3.2 4 1.1 -1	$(3.2+4i) + (1.1-1i) = (4.3+3i)$ $(3.2+4i) - (1.1-1i) = (2.1+5i)$ $(3.2+4i) * (1.1-1i) = (7.5+1.2i)$ $(3.2+4i) / (1.1-1i) = (-0.22+3.4i)$	$(3.2+4i) + (1.1-1i) = (4.3+3i)$ $(3.2+4i) - (1.1-1i) = (2.1+5i)$ $(3.2+4i) * (1.1-1i) = (7.5+1.2i)$ $(3.2+4i) / (1.1-1i) = (-0.22+3.4i)$	✓
✓	5.5 2 3 -1.5	$(5.5+2i) + (3-1.5i) = (8.5+0.5i)$ $(5.5+2i) - (3-1.5i) = (2.5+3.5i)$ $(5.5+2i) * (3-1.5i) = (20-2.2i)$ $(5.5+2i) / (3-1.5i) = (1.2+1.3i)$	$(5.5+2i) + (3-1.5i) = (8.5+0.5i)$ $(5.5+2i) - (3-1.5i) = (2.5+3.5i)$ $(5.5+2i) * (3-1.5i) = (20-2.2i)$ $(5.5+2i) / (3-1.5i) = (1.2+1.3i)$	✓

Passed all tests! ✓

Figure 44

Bài tập 2.6. Giả thuyết Collatz: bắt đầu từ số dương n bất kỳ, nếu n chẵn thì chia 2, nếu lẻ thì nhân 3 cộng 1, giả thuyết cho rằng ta luôn đi đến $n=1$. Hãy viết chương trình mô phỏng lại quá trình biến đổi để kiểm chứng giả thuyết với giá trị của n nhập từ bàn phím.

```

1  /*
2   Bài 2.6. Giả thuyết Collatz: bắt đầu từ số dương n bất kỳ,
3   nếu n chẵn thì chia 2, nếu lẻ thì nhân 3 cộng 1, giả thuyết
4   cho rằng ta luôn đi đến n = 1. Hãy viết chương trình mô phỏng
5   lại quá trình biến đổi để kiểm chứng giả thuyết với giá trị
6   của n nhập từ bàn phím.
7 */
8 //Le Dinh Tri Tue - 0909 - 732833
9 #include <stdio.h>
10
11 void print(int n) {
12     printf("n=%d\n", n);
13 }
14 //Le Dinh Tri Tue - 0909 - 732833
15 //Nhân 3 cộng 11
16 int mul3plus1(int n) {
17     return n * 3 + 1;
18 }
19 //Le Dinh Tri Tue - 0909 - 732833
20 //Chia 2
21 int div2(int n) {
22     return n / 2;

```

Figure 45

```
21     int n1vz(int n) {
22         return n / 2;
23     }
24
25 // khai báo các tham số cho các con trả hàm odd, even và output
26 void simulate(int n, int (*odd)(int), int (*even)(int), void (*output)(int)) {
27     (*output)(n);
28     if (n == 1) return;
29     if (n % 2 == 0) {
30         n = (*even)(n);
31     } else {
32         n = (*odd)(n);
33     }
34     simulate(n, odd, even, output);
35 }
36 //Le Dinh Tri Tue - 0909 - 732833
37 int main() {
38     int (*odd)(int) = &mul3plus1;
39     int (*even)(int) = &div2;
40
41     int n;
42     scanf("%d" &n).
```

Figure 46

```
36 //Le Dinh Tri Tue - 0909 - 732833
37 int main() {
38     int (*odd)(int) = &mul3plus1;
39     int (*even)(int) = &div2;
40
41     int n;
42     scanf("%d", &n);
43     simulate(n, odd, even, print);
44
45     return 0;
46 }
47
```

Precheck

Check

	Input	Expected	Got		
✓	19	n=19 n=58 n=29 n=88 n=44 n=22 n=11 n=34 n=17 n=52 n=26 n=13 n=40 n=20	n=19 n=58 n=29 n=88 n=44 n=22 n=11 n=34 n=17 n=52 n=26 n=13 n=40 n=20	✓	

Figure 47

		n=20 n=10 n=5 n=16 n=8 n=4 n=2 n=1	n=20 n=10 n=5 n=16 n=8 n=4 n=2 n=1		
✓	33	n=33 n=100 n=50 n=25 n=76 n=38 n=19 n=58 n=29 n=88 n=44 n=22 n=11 n=34 n=17 n=52 n=26 n=13 n=40 n=20 n=10 n=5 n=16 n=8 n=4 n=2 n=1	n=33 n=100 n=50 n=25 n=76 n=38 n=19 n=58 n=29 n=88 n=44 n=22 n=11 n=34 n=17 n=52 n=26 n=13 n=40 n=20 n=10 n=5 n=16 n=8 n=4 n=2 n=1	✓	

Passed all tests! ✓

Figure 48

Bài tập 2.7. Viết hàm tính tổng các phần tử trong hai mảng. Yêu cầu sử dụng function template để cho phép hàm làm việc với các mảng số nguyên lẫn số thực.

```

1 /*
2 Bai 2.7. Viết hàm tính tổng các phần tử trong hai mảng.
3 Yêu cầu sử dụng function template để cho phép hàm làm việc với các mảng số nguyên lẫn số thực.
4 //Le Dinh Tri Tue - 0909 - 732833
5 */
6
7 #include <iostream>
8 using namespace std;
9 //Le Dinh Tri Tue - 0909 - 732833
10 //# viết hàm arr_sum
11 ****
12 # YOUR CODE HERE #
13 ****
14 //Le Dinh Tri Tue - 0909 - 732833
15 template <typename T>
16 //Hàm tính tổng cả số nguyên lẫn số thực viết bằng template
17 T arr_sum(T arr1[], int size1, T arr2[], int size2) {
18     T sum = 0;
19     for (int i = 0; i < size1; ++i) {
20         sum = sum + arr1[i];
21     }
22     for (int i = 0; i < size2; ++i) {

```

Figure 49

```

21     }
22     for (int i = 0; i < size2; ++i) {
23         sum = sum + arr2[i];
24     }
25     return sum;
26 }
27 //Le Dinh Tri Tue - 0909 - 732833
28 int main() {
29     int val;
30     cin >> val;
31     // Hàm arr_sum xử lý với số ngunguyên
32     //Le Dinh Tri Tue - 0909 - 732833
33     {
34         int a[] = {3, 2, 0, val};
35         int b[] = {5, 6, 1, 2, 7};
36         cout << arr_sum(a, 4, b, 5) << endl;
37     }
38     //Hàm arr_sum xử lý với số thực
39     //Le Dinh Tri Tue - 0909 - 732833

```

Figure 50

```
32 //Le Dinh Tri Tue - 0909 - 732833
33 {
34     int a[] = {3, 2, 0, val};
35     int b[] = {5, 6, 1, 2, 7};
36     cout << arr_sum(a, 4, b, 5) << endl;
37 }
38 //Hàm arr_sum xử lý với số thực
39 //Le Dinh Tri Tue - 0909 - 732833
40 {
41     double a[] = {3.0, 2, 0, val * 1.0};
42     double b[] = {5, 6.1, 1, 2.3, 7};
43     cout << arr_sum(a, 4, b, 5) << endl;
44 }
45
46 return 0;
47 }
```

Precheck

Check

	Input	Expected	Got	
✓	5	31 31.4	31 31.4	✓
✓	17	43 43.4	43 43.4	✓

Passed all tests! ✓

Figure 51

Bài tập 2.8. Viết hàm so sánh cho thuật toán sắp xếp.

```
1  /*
2  Bài 2.8. Viết hàm so sánh cho thuật toán sắp xếp.
3  //Le Dinh Tri Tue - 0909 - 732833
4  */
5  #include <iostream>
6  #include <vector>
7  #include <algorithm>
8  #include <numeric>
9  //Le Dinh Tri Tue - 0909 - 732833
10 using namespace std;
11
12 int main() {
13     int val1, val2;
14     cin >> val1 >> val2;
15     vector< vector<int> > a = {
16         {1, 3, 7},
17         {2, 3, 4, val1},
18         {9, 8, 15},
19         {10, val2},
20     };
21     //Le Dinh Tri Tue - 0909 - 732833
22     // Sắp xếp các vector trong a theo tổng các phần tử giảm dần
```

Figure 52

```

20   };
21   //Le Dinh Tri Tue - 0909 - 732833
22   // Sắp xếp các vector trong a theo tổng các phần tử giảm dần
23   sort(a.begin(), a.end(), [](const vector<int>& v1, const vector<int>& v2) {
24     int sum1 = accumulate(v1.begin(), v1.end(), 0);
25     int sum2 = accumulate(v2.begin(), v2.end(), 0);
26     return sum1 > sum2;
27   });
28   //Le Dinh Tri Tue - 0909 - 732833
29   // In ra kết quả
30   for (const auto &v : a) {
31     for (int it : v) {
32       cout << it << ' ';
33     }
34     cout << endl;
35   }
36
37   return 0;
38 }
39

```

Precheck Check

	Input	Expected	Got	
✓	-10 -5	9 8 15 1 3 7 10 -5 2 3 4 -10	9 8 15 1 3 7 10 -5 2 3 4 -10	✓
✓	100 -100	2 3 4 100 9 8 15 1 3 7 10 -100	2 3 4 100 9 8 15 1 3 7 10 -100	✓

Passed all tests! ✓

Figure 53

Bài tập 2.9. Dưới đây cung cấp đoạn code đơn giản để tính hàm sigmoid theo công thức trực tiếp. Hãy viết hàm tính xấp xỉ sigmoid(x) đến độ chính xác 10–6 và có tốc độ nhanh hơn ít nhất 30% so với code đơn giản.

```

1 /*
2 Bài 9: Dưới đây cung cấp đoạn code đơn giản để tính hàm sigmoid theo công thức trực tiếp.
3 Hãy viết hàm tính xấp xỉ sigmoid(x) đến độ chính xác 10-6 và có tốc độ nhanh hơn ít nhất
4 30% so với code đơn giản.
5 Gợi ý: sử dụng kỹ thuật "chuẩn bị trước" như trong slide.
6 */
7 //Le Dinh Tri Tue - 0909 - 732833
8
9 #include <vector>
10 #include <algorithm>
11 #include <cmath>
12 #include <ctime>
13 #include <algorithm>
14 #include <cstdio>
15
16 using namespace std;
17 //Le Dinh Tri Tue - 0909 - 732833
18 const int LIMIT = 100;
19 const int NUM_ITER = 100000;
20 const int NUM_INPUTS = NUM_ITER * 100;
21
22 double sigmoid_slow(double x) {
```

Figure 54

```

22 double sigmoid_slow(double x) {
23     return 1.0 / (1.0 + exp(-x));
24 }
25
26 double x[NUM_INPUTS];
27 //Le Dinh Tri Tue - 0909 - 732833
28 void prepare_input() {
29     const int PRECISION = 1000000;
30     const double RANGE = LIMIT / 20.0;
31     for (int i = 0; i < NUM_INPUTS; ++i) {
32         x[i] = RANGE * (rand() % PRECISION - rand() % PRECISION) / PRECISION;
33     }
34     return;
35 }
36
37 //# BEGIN fast code
38
39 //# khai báo các biến phụ trợ cần thiết
40 *****
41 const int TABLE_SIZE = 100000 ;
42 double sigmoid_table[TABLE_SIZE + 1];
43 *****
```

Figure 55

```

60     if (x < -LIMIT) return 0;
61     else if (x > LIMIT) return 1;
62     else {
63         double delta1 = 2.0 * LIMIT / TABLE_SIZE;
64         int i = (x + LIMIT) / delta1;
65         return (sigmoid_table[i+1] - sigmoid_table[i]) / delta1 * (x + LIMIT - i * delta1) + sigmoid_table[i];
66     }
67     *****/
68 }
69 //Le Dinh Tri Tue - 0909 - 732833
70 //# END fast code
71
72 double benchmark(double (*calc)(double), vector<double> &result) {
73     const int NUM_TEST = 20;
74
75     double taken = 0;
76     result = vector<double>();
77     result.reserve(NUM_ITER);
78
79     int input_id = 0;
80     clock_t start = clock();
81     for (int t = 0; t < NUM_TEST; ++t) {

```

Figure 56

```

79     int input_id = 0;
80     clock_t start = clock();
81     for (int t = 0; t < NUM_TEST; ++t) {
82         double sum = 0;
83         for (int i = 0; i < NUM_ITER; ++i) {
84             double v = fabs(calc(x[input_id]));
85             sum += v;
86             if (t == 0) result.push_back(v);
87             if ((++input_id) == NUM_INPUTS) input_id = 0;
88         }
89     }
90     clock_t finish = clock();
91     taken = (double)(finish - start);
92     //# printf("Time: %.9f\n", taken / CLOCKS_PER_SEC);
93     return taken;
94 }
95 //Le Dinh Tri Tue - 0909 - 732833
96 bool is_correct(const vector<double> &a, const vector<double> &b) {
97     const double EPS = 1e-6;
98
99     if (a.size() != b.size()) return false;
100    for (int i = 0; i < (int)a.size(); ++i) {

```

Figure 57

```
96 v bool is_correct(const vector<double> &a, const vector<double> &b) {
97     const double EPS = 1e-6;
98
99     if (a.size() != b.size()) return false;
100    for (int i = 0; i < (int)a.size(); ++i) {
101        if (fabs(a[i] - b[i]) > EPS) {
102            return false;
103        }
104    }
105    return true;
106}
107 //Le Dinh Tri Tue - 0909 - 732833
108 int main() {
109     prepare_input();
110     precalc();
111
112     vector<double> a, b;
113     double slow = benchmark(sigmoid_slow, a);
114     double fast = benchmark(sigmoid_fast, b);
115
116     double xval;
117     scanf("%lf", &xval);
```

Figure 58

```

107 //Le Dinh Tri Tue - 0909 - 732833
108 int main() {
109     prepare_input();
110     precalc();
111
112     vector<double> a, b;
113     double slow = benchmark(sigmoid_slow, a);
114     double fast = benchmark(sigmoid_fast, b);
115
116     double xval;
117     scanf("%lf", &xval);
118     printf("%.2f\n", sigmoid_fast(xval));
119     if (is_correct(a, b) && (slow/fast > 1.3)) {
120         printf("Correct answer! Your code is faster at least 30%!\n");
121     } else {
122         printf("Wrong answer or your code is not fast enough!\n");
123     }
124
125     return 0;
126 }
127

```

Check

	Input	Expected	Got	
✓	1.5	0.82 Correct answer! Your code is faster at least 30%	0.82 Correct answer! Your code is faster at least 30%	✓
✓	2.15	0.90 Correct answer! Your code is faster at least 30%	0.90 Correct answer! Your code is faster at least 30%	✓

Passed all tests! ✓

Figure 59

Bài tập 2.10 (BONUS). Dưới đây cung cấp đoạn code đơn giản để tính tích của hai ma trận cỡ $N \times N$ theo công thức trực tiếp. Hãy viết hàm tính tích hai ma trận nhưng có tốc độ nhanh hơn ít nhất 10% so với code đơn giản. Gợi ý: hãy để ý đến thứ tự truy cập các phần tử trong ma trận, tối ưu cache hoặc sử dụng thuật toán tốt hơn $O(N^3)$.

CODE:

/*

Bài 10: Dưới đây cung cấp đoạn code đơn giản để tính tích của hai ma trận cỡ $N \times N$ theo công thức trực tiếp.

Hãy viết hàm tính tích hai ma trận nhưng có tốc độ nhanh hơn ít nhất 10% so với code đơn giản.

Gợi ý: hãy để ý đến thứ tự truy cập các phần tử trong ma trận, tối ưu cache hoặc sử dụng thuật toán tốt hơn $O(N^3)$

// Le Dinh Tri Tue - 0909 - 732833

```
*/
```

```
#include <iostream>
#include <cstring>
#include <ctime>

using namespace std;

const int N = 128;
// Le Dinh Tri Tue - 0909 - 732833

struct Matrix {
    unsigned int mat[N][N];

    Matrix() {
        memset(mat, 0, sizeof mat);
    }
};

// Le Dinh Tri Tue - 0909 - 732833

bool operator == (const Matrix &a, const Matrix &b) {
    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < N; ++j) {
            if (a.mat[i][j] != b.mat[i][j]) return false;
        }
    }
    return true;
}

// Le Dinh Tri Tue - 0909 - 732833

Matrix multiply_naive(const Matrix &a, const Matrix &b) {
    Matrix c;
    for (int i = 0; i < N; ++i) {
```

```
for (int j = 0; j < N; ++j) {  
    for (int k = 0; k < N; ++k) {  
        c.mat[i][j] += a.mat[i][k] * b.mat[k][j];  
    }  
}  
  
for (int i = 0; i < N; ++i) {  
    for (int j = 0; j < N; ++j) {  
        for (int k = 0; k < N; ++k) {  
            c.mat[i][j] += 0;  
        }  
    }  
}  
return c;  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
Matrix operator + (const Matrix &a, const Matrix &b) {  
    Matrix result;  
    for (int i = 0; i < N; ++i) {  
        for (int j = 0; j < N; ++j) {  
            result.mat[i][j] = a.mat[i][j] + b.mat[i][j];  
        }  
    }  
    return result;  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
Matrix operator - (const Matrix &a, const Matrix &b) {  
    Matrix result;  
    for (int i = 0; i < N; ++i) {  
        for (int j = 0; j < N; ++j) {  
            result.mat[i][j] = a.mat[i][j] - b.mat[i][j];  
        }  
    }  
    return result;  
}
```

```
    result.mat[i][j] = a.mat[i][j] - b.mat[i][j];  
}  
}  
return result;  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
Matrix multiply_fast(const Matrix &a, const Matrix &b) {  
    Matrix c;  
    for (int i = 0; i < N; ++i) {  
        for (int j = 0; j < N; ++j) {  
            int sum = 0;  
            for (int k = 0; k < N; ++k) {  
                sum += a.mat[i][k] * b.mat[k][j];  
            }  
            c.mat[i][j] = sum;  
        }  
    }  
    return c;  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
Matrix gen_random_matrix() {  
    Matrix a;  
    for (int i = 0; i < N; ++i) {  
        for (int j = 0; j < N; ++j) {  
            a.mat[i][j] = rand();  
        }  
    }  
    return a;  
}
```

Matrix base;

// Le Dinh Tri Tue - 0909 - 732833

```
double benchmark(Matrix (*multiply) (const Matrix&, const Matrix&), Matrix &result) {
```

```
    const int NUM_TEST = 10;
```

```
    const int NUM_ITER = 64;
```

```
    Matrix a = base;
```

```
    result = a;
```

```
    double taken = 0;
```

```
    for (int t = 0; t < NUM_TEST; ++t) {
```

```
        clock_t start = clock();
```

```
        for (int i = 0; i < NUM_ITER; ++i) {
```

```
            a = multiply(a, result);
```

```
            result = multiply(result, a);
```

```
        }
```

```
        clock_t finish = clock();
```

```
        taken += (double)(finish - start);
```

```
}
```

```
taken /= NUM_TEST;
```

```
printf("Time: %.9f\n", taken / CLOCKS_PER_SEC);
```

```
return taken;
```

```
}
```

// Le Dinh Tri Tue - 0909 - 732833

```
int main() {
```

```
    base = gen_random_matrix();
```

```
    Matrix a, b;
```

```
    printf("Slow version\n");
```

```
double slow = benchmark(multiply_naive, a);
printf("Fast version\n");
double fast = benchmark(multiply_fast, b);
if (a == b) {
    printf("Correct answer! Your code is %.2f%% faster\n", slow / fast * 100.0);
} else {
    printf("Wrong answer!\n");
}
return 0;
}

// Le Dinh Tri Tue - 0909 - 732833
```

```
mpCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh:
empCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
input = 2
2 1
5 3
-1 4
3 6
ouput = 1 5
8 9
1 14
4 38
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh:
empCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
input = 3
5 1 7
1 2 3
4 9 8
3 -1 8
2 5 7
4 9 6
ouput = 8 0 15
3 7 10
8 18 14
45 63 89
19 36 40
62 113 143
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh:
```

Hình ảnh 34 – Bài 1.10

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Tri
empCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
input = 7
0 2 0 2 3 3 0
1 3 5 5 5 3 3
1 2 1 5 2 3 4
2 3 1 0 5 1 3
4 3 2 4 2 1 1
4 1 1 1 4 5 2
0 2 2 2 3 1 2
5 2 1 3 0 3 2
1 1 3 1 2 2 3
4 3 1 5 2 3 3
3 1 4 4 1 3 2
2 2 5 4 1 3 4
1 3 4 1 2 0 4
2 1 2 3 0 0 5
output = 5 4 1 5 3 6 2
2 4 8 6 7 5 6
5 5 2 10 4 6 7
5 4 5 4 6 4 5
6 5 7 8 3 4 5
5 4 5 2 6 5 6
2 3 4 5 3 1 7
17 19 41 25 15 19 34
62 47 78 83 32 54 83
41 29 58 53 19 31 61
34 26 47 44 15 30 55
50 29 47 53 18 42 48
45 38 56 49 19 32 62
27 21 39 39 15 25 42
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Tri
```

Hình ảnh 35 – Bài 1.10

```
empCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
input = 10
53 44 29 68 32 19 68 24 84 94
33 53 67 1 78 74 55 12 83 65
11 46 70 60 47 24 43 61 26 48
61 88 7 39 4 92 64 45 61 64
78 52 25 92 18 27 65 46 58 54
42 94 22 0 72 77 68 66 54 65
9 64 67 46 47 19 64 77 87 54
27 96 84 42 15 29 95 35 26 72
15 24 34 46 34 9 21 66 32 52
28 68 78 77 61 58 71 42 12 99
22 93 39 24 33 81 77 75 19 88
11 97 67 44 42 76 17 83 50 99
11 19 74 69 27 38 86 79 15 0
6 14 52 36 20 22 99 32 5 87
40 70 35 74 82 79 51 53 85 61
45 9 15 39 43 95 88 9 56 99
37 53 59 68 89 33 59 11 93 53
76 53 26 70 39 67 15 14 70 48
63 86 4 44 77 44 20 68 58 80
77 48 75 2 40 83 41 50 82 23
ouput = 75 137 68 92 65 100 145 99 103 182
44 150 134 45 120 150 72 95 133 164
22 65 144 129 74 62 129 140 41 48
67 102 59 75 24 114 163 77 66 151
118 122 60 166 100 106 116 99 143 115
87 103 37 39 115 172 156 75 110 164
46 117 126 114 136 52 123 88 180 107
38729
18763 29027 23508 21882 24115 29492 29341 24285 25016 35322
24484 34041 24058 26743 30527 38822 25919 25793 35666 37604
22509 29970 24095 27208 27943 30250 25051 25812 30481 31168
18088 27900 28864 24619 25210 30139 28012 25932 28379 29771
14822 17333 15409 15845 15503 19723 16568 15089 18312 18700
21932 28325 31183 27088 27763 36325 34976 27260 32277 34374
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh>
```

Hình ảnh 36 – Bài 1.10

Bài tập 2.11. Cho 2 đa thức $A(x)$ và $B(x)$ tương ứng có bậc N và M . Hãy tính ma trận tích $C(x) = A(x) * B(x)$ có bậc $N+M-1$. Input: Gồm 2 dòng biểu diễn các đa thức $A(x)$ và $B(x)$, mỗi dòng Số đầu tiên N là bậc của đa thức; $N+1$ số nguyên tiếp theo, số thứ i là hệ số của x^{i-1} . Output: Một số nguyên duy nhất là XOR của các hệ số của đa thức $C(x)$.

CODE:

```
/*
```

Bài 11: Cho 2 đa thức $A(x)$ và $B(x)$ tương ứng có bậc N và M

Hãy tính ma trận tích $C(x) = A(x) * B(x)$ có bậc $N+M-1$

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
*/
```

```
#include <bits/stdc++.h>
```

```
void find_index(int a[], int b[], int n, int m, int c[]) {
```

```
    for (int i = 0; i < n + m + 1; i++) {
```

```
        for (int j = 0; j < n + 1; j++) {
```

```
            if (i - j >= 0 and i - j < m + 1) {
```

```
                c[i] += a[j] * b[i-j];
```

```
            }
```

```
        }
```

```
}
```

```
}
```

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
int cal_XOR(int c[], int n) {
```

```
    if (n > 0) {
```

```
        int result = c[0];
```

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

```
            result = result ^ c[i];
```

```
        }
```

```
        return result;
```

```
}
```

```

return -1;
}

// Le Dinh Tri Tue - 0909 - 732833

int main() {
    int n;
    std::cin>>n;
    int a[n + 1];
    for (int i = 0; i < n + 1; i++) {
        std::cin>>a[i];
    }
    int m;
    std::cin>>m;
    int b[m + 1];
    for (int i = 0; i < m + 1; i++) {
        std::cin>>b[i];
    }
    int c[m+n+1];
    memset(c,0,sizeof c);
    find_index(a, b, n, m, c);
    std::cout<<cal_XOR(c, n + m);
}
}

// Le Dinh Tri Tue - 0909 - 732833

```

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
o tesst } ; if ($?) { g++ tesst.cpp
9 21 62 27 90 59 63 26 48 26 72
8 36 11 68 67 29 82 30 62 23
5196
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
o tesst } ; if ($?) { g++ tesst.cpp
99 67 35 29 2 22 58 69 67 93 56 11 42 29 73 21 19 84 37 98 24 15 70 13 26 91 80 56 73 62 70 96 81 5 25 84 27 36 5 46 29 13 57 24 95 82 45 14 67 34 64 43 50 87 8 76 78 88 84 3 51 54 99 32 60 76 68 39 12
26 86 94 39 95 70 34 78 67 1 97 2 17 92 52 56 1 80 86 41 65 89 44 19 40 29 31 17 97 71 81 75
98 9 27 67 56 97 53 86 65 6 83 19 24 28 71 32 29 3 19 70 68 8 15 40 49 96 23 18 45 46 51 21 55 79 88 64 28 41 50 93 0 34 64 24 14 87 56 43 91 27 65 59 36 32 51 37 28 75 7 74 21 58 95 29 37 35 93 18 28
43 11 28 29 76 4 43 63 13 38 6 40 4 18 28 88 69 17 17 96 24 43 70 83 90 99 72 25 44 90 5
125199
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
o tesst } ; if ($?) { g++ tesst.cpp
000 20 53 85 60 93 12 54 07 7 55 4 49 11 22 38 99 42 46 59 40 32 13 10 5 1 61 20 70 5 20 36 44 26 22 55 8 16 82 24 27 62 24 8 26 53 99 70 50 69 71 21 21 20 32 24 69 62 20 21 20 26 50 72 17 68 99

```

Figure 60

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT\" ; if (?) { g++ tesst.cpp
o tesst } ; if (?) { \ntest }

999 39 54 86 69 82 42 64 97 7 55 4 48 11 22 28 99 43 46 68 40 22 11 10 5 1 61 39 78 5 20 36 44 26 22 65 8 16 82 58 24 37 62 24 0 36 52 99 79 50 68 71 73 31 81 30 33 94 60 63 99 81 99 96 59 73 13 68 9
95 26 66 84 48 98 84 76 42 36 7 45 56 79 18 87 12 48 72 59 9 36 18 42 87 6 1 13 72 21 55 19 99 21 4 39 11 48 67 5 28 27 50 84 58 20 24 22 69 96 81 38 84 92 72 72 50 25 85 22 99 40 42 98 13 98 96 24
9 81 19 36 32 55 94 4 79 69 73 76 50 55 60 42 79 84 93 5 21 67 4 13 61 54 26 59 44 2 2 6 84 21 42 68 28 89 72 8 58 98 36 8 53 48 3 33 33 48 90 54 67 46 68 29 0 46 88 97 49 90 3 33 63 93 52 92 86 25
96 75 88 57 29 36 68 14 21 60 4 28 27 50 48 52 2 94 97 99 43 39 2 28 3 0 81 47 38 59 51 35 34 39 92 15 27 4 29 49 64 89 29 43 35 77 0 38 71 49 89 67 88 92 95 43 44 29 98 82 40 41 69 26 32 61 42 68 1
3 61 81 9 96 26 96 67 77 34 90 26 24 57 14 68 5 58 12 86 0 46 26 94 16 52 78 29 46 90 47 70 51 80 31 93 57 27 12 86 14 55 12 90 12 79 10 69 89 74 55 41 20 33 87 88 38 66 70 84 56 17 6 60 49 37 5 59
18 45 83 73 58 73 37 89 83 7 78 57 14 71 29 0 58 18 38 25 88 74 33 57 81 93 58 70 99 17 39 69 63 22 94 73 47 31 62 82 90 92 91 51 57 21 57 94 71 47 51 31 21 37 40 54 30 98 25 81 16 2 31 39 96 4 38
0 18 21 70 62 12 79 77 85 36 4 76 83 7 59 57 44 99 11 27 50 36 60 18 5 63 49 44 11 5 34 91 75 55 14 89 68 93 18 8 52 22 82 17 30 93 74 26 93 86 53 43 74 14 13 79 77 62 75 88 19 10 32 94 17 46 35 37 5
53 43 73 28 91 10 18 17 36 63 55 99 58 30 4 71 61 33 85 89 73 4 51 5 50 68 3 85 6 95 39 49 20 67 26 63 77 96 81 65 60 36 53 42 32 96 79 21 70 84 72 27 34 40 83 72 98 30 73 14
1 22 47 24 82 35 32 4 54 43 86 48 79 59 62 62 83 41 48 23 24 72 22 54 35 21 57 65 47 71 76 69 18 1 3 53 33 7 59 28 6 97 20 84 8 4 34 98 91 76 98 15 52 71 89 59 6 10 16 24 9 39 0 78 9 53 81 14 38 89 2
67 47 23 87 31 32 22 81 75 50 79 94 54 50 31 13 57 94 81 81 3 20 33 82 81 87 15 96 25 4 22 92 51 97 32 34 81 6 15 57 8 95 99 62 97 83 76 54 77 9 87 32 82 21 66 63 60 82 11 85 86 85 30 90 83 14 76 16
92 25 28 39 25 90 36 60 18 43 37 28 28 82 21 10 55 88 25 15 70 37 53 8 22 83 50 57 97 27 26 69 71 51 49 10 28 39 88 10 93 77 90 76 99 52 31 87 77 99 57 66 52 17 41 35 68 98 84 95 76 5 66 28 54 28 8
78 97 55 72 74 45 0 25 97 83 12 27 82 21 93 34 39 24 21 59 85 57 54 61 62 72 41 16 52 50 62 82 99 17 54 73 15 6 51 64 90 63 91 72 37 37 59 28 71 80 87 56 90 41 70 52 65 65 11 69 17 61 83 51 12 0 6 38 0
64 89 32 54 4 75 79 41 12 38 69 36 70 56 44 60 49 14 65 14 26 86 83 39 69 35 52 21 93 90 89 9 31 73 64 35 48 95 77 13 33 98 49 55 55 93 68 56
1146480
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT>

```

Figure 61

Bài tập 2.12. Hôm nay, cô giáo giao cho An một câu hỏi hóc búa. Cô cho một danh sách với mỗi phần tử có dạng `<key, value>` và yêu cầu An sắp xếp danh sách đó giảm dần theo giá trị `value`. Nếu 2 phần tử có `value` giống nhau thì sắp xếp giảm dần theo `key`. Hãy viết một chương trình sử dụng hàm `nặc danh` để giúp An làm bài tập. Input: Danh sách đầu vào. Mỗi dòng ghi một cặp giá trị `key, value` cách nhau bởi dấu cách ($|key| \leq 109$, $|value| \leq 109$). Output: In danh sách đã được sắp xếp theo yêu cầu. Mỗi dòng ghi một cặp giá trị `key, value` cách nhau bởi dấu cách.

CODE:

```
/*

```

Bài 12: Hôm nay, cô giáo giao cho An một câu hỏi hóc búa. Cô cho một danh sách với mỗi phần tử có dạng `<key, value>` và yêu cầu An sắp xếp danh sách đó giảm dần theo giá trị `value`. Nếu 2 phần tử có `value` giống nhau thì sắp xếp giảm dần theo `key`.

Hãy viết một chương trình sử dụng hàm `nặc danh` để giúp An làm bài tập.

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
*/

```

```
#include <bits/stdc++.h>
```

```
bool customCompare(const std::pair<int, int> &a, const std::pair<int, int> &b) {
```

```
    if (a.second == b.second) {
```

```
    return a.first > b.first;  
}  
  
    return a.second > b.second;  
}  
  
  
int main() {  
  
    std::vector<std::pair<int, int>> myList;  
  
    int n;  
  
    std::cin>>n;  
  
    for (int i = 0; i < n; i++) {  
  
        int a, b;  
  
        std::cin>>a>>b;  
  
        myList.push_back(std::make_pair(a, b));  
    }  
  
    std::sort(myList.begin(), myList.end(), customCompare);  
  
  
    for (const auto& pair : myList) {  
  
        std::cout<< pair.first <<" "<< pair.second <<"\n";  
    }  
  
    return 0;  
}  
  
// Le Dinh Tri Tue - 0909 – 732833
```

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT\" ; i
f ($?) { g++ tesst.cpp -o tesst } ; if ($?) { .\tesst }
2 3
4 8
9 1
-999999
4 8
2 3
9 1
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT\" ; i
f (?) { g++ tesst.cpp -o tesst } ; if (?) { .\tesst }
4 -8 -6 2 6 -4 -6 7 -9 5 3 1 -2 1 -3 -1 2 1 -8 3 8 10 -8 -9 0 -4 -4 -4 7 9 0 5 -2 -6 5 -10 -7 -4 0 9 2 6 5 5 -4 9 2 -4 -4 0 1 -5 -7 -3 -2 8 1 5 -9 9 -7 -2 -2 -
6 0 -8 -1 -2 -3 -8 -1 9 1 2 6 -2 6 10 0 -3 3 2 -7 -10 -1 1 10 10 -6 -10 5 1 6 -1 -10 -9 -9 -1 -1 -5 6 2 -2 -6 -6 -2 -8 -8 -9 7 -9 10 -6 1 -5 8 -9 4 5 -5 -10 1 0 -7 9 5 -9 -10 -9
-10 5 7 8 3 1 -2 -4 -7 -8 -2 6 -6 -10 5 -1 -1 -10 -8 -9 -8 2 9 1 1 -6 -2 9 -3 -7 -9 2 10 3 10 2 -10 0 -10 5 -9 -2 -3 8 4 4 6 4 1 1 6 -8 -10 -7 -9 10 -1 -3 -6 2 0 -5 2
-999999
3 10
2 10
1 10
2 9
-7 9
7 8
3 8
-1 8
-4 7
4 6
2 6
2 6
1 6
1 6
-2 6
-2 6

```

Figure 62

```
1 6
1 6
-2 6
-2 6
-5 6
10 5
6 5
4 5
-6 5
-10 5
8 4
5 3
-3 3
9 2
9 2
-1 2
-5 2
9 1
8 1
4 1
3 1
1 1
0 1
10 0
9 0
2 0
1 0
-4 0
-6 0
```

Tuan12-Tuet DT Debug ⌂ ⌂ ⌂ ⌂ 0

Figure 63

```
-6 0
-9 0
10 -1
5 -1
2 -1
-1 -1
-1 -1
-8 -1
-8 -1
-10 -1
5 -2
2 -2
1 -2
-2 -2
-3 -2
-6 -2
9 -3
1 -3
0 -3
-2 -3
-2 -3
5 -4
-2 -4
-4 -4
-4 -4
1 -5
10 -6
10 -6
-3 -6
-4 -6
-6 -6
```

Figure 64

```

-6 -6
9 -7
2 -7
-5 -7
-10 -7
10 -8
4 -8
1 -8
-2 -8
-7 -8
-9 -8
-10 -8
8 -9
7 -9
7 -9
5 -9
5 -9
5 -9
-7 -9
-7 -9
-8 -9
-9 -9
-10 -9
2 -10
0 -10
-1 -10
-5 -10
-6 -10
-8 -10

```

Figure 65

Bài tập 2.13. Số nguyên lớn là các số nguyên có giá trị rất lớn và không thể biểu diễn bằng các kiểu dữ liệu nguyên cơ bản. Để biểu diễn số nguyên lớn, ta có thể dùng kiểu struct như sau: struct bigNum { char sign; char num[101]; }; Nhiệm vụ các bạn là đa năng hóa các toán tử để thực hiện các phép toán số học với kiểu dữ liệu số nguyên lớn vừa định nghĩa ở trên. Input: Dữ liệu vào gồm hai dòng mô tả hai số nguyên lớn a và b , mỗi dòng chứa 1 chuỗi ký tự mô tả 1 số nguyên lớn không vượt quá 10100 . Chữ số đầu của mỗi chuỗi ký tự sẽ thể hiện dấu của số đó: 0 là âm, 1 là dương. Các chữ số sau thể hiện giá trị của số đó. Output: In ra giá trị của biểu thức $ab - 3a + 4b$. Kết quả in ra một số nguyên lớn dưới dạng chuỗi ký tự có định dạng như mô tả trong dữ liệu vào.

CODE:

```
/*
```

Bài 13: Số nguyên lớn là các số nguyên có giá trị rất lớn và không thể biểu diễn bằng các kiểu dữ liệu nguyên cơ bản. Để biểu diễn số nguyên lớn, ta có thể dùng kiểu struct như sau:

```
struct bigNum{
    char sign;
    char num[101];
};
```

Nhiệm vụ các bạn là đa năng hóa các toán tử để thực hiện các phép toán số học với kiểu dữ liệu số nguyên lớn vừa định nghĩa ở trên.

Le Dinh Tri Tue - 0909 - 732833

```
*/
```

```
#include<bits/stdc++.h>
using namespace std;

struct bigNum {
    char sign;
    char num[101];
};

// Nhập vào hai số nguyên lớn
// Le Dinh Tri Tue - 0909 - 732833
void input(bigNum &num1, bigNum &num2){
    string tmp;
    cin >> tmp;
    num1.sign = tmp[0];
    int lens1 = tmp.length() - 1;
    for (int i = 0; i < lens1; i++){
        num1.num[100-lens1+i+1] = tmp[i+1];
    }
}
```

```
for (int i = 0; i < 100-lens1+1; i++){
    num1.num[i] = '0';
}

cin >> tmp;
num2.sign = tmp[0];
int lens2 = tmp.length() - 1;
for (int i = 0; i < lens2; i++){
    num2.num[100-lens2+i+1] = tmp[i+1];
}
for (int i = 0; i < 100-lens2+1; i++){
    num2.num[i] = '0';
}
}

// cộng hai số nguyên dương lớn
// Le Dinh Tri Tue - 0909 - 732833
void add(char res[], char *num1, char *num2){
    int c = 0;
    for(int i=100; i>=0; i--){
        int tmp = (int)num1[i] - 48 + (int)num2[i] - 48 + c;
        c = tmp / 10;
        res[i] = tmp % 10 + 48;
    }
}

// trừ hai số nguyên dương lớn, num1 > num2
// Le Dinh Tri Tue - 0909 - 732833
void sub(char res[], char *num1, char* num2){
    int c = 0;
    for (int i = 100; i >= 0; i--){
```

```
int tmp1 = (int)num1[i] - 48;  
int tmp2 = (int)num2[i] - 48;  
  
if (tmp1 >= tmp2 + c){  
    res[i] = tmp1 - tmp2 - c + 48;  
    c = 0;  
}  
else {  
    tmp1 = tmp1 + 10;  
    res[i] = tmp1 - tmp2 - c + 48;  
    c = 1;  
}  
}  
  
}  
  
// nhân hai số nguyên lớn  
// Le Dinh Tri Tue - 0909 - 732833  
void multi(char res[], char *num1, char *num2){  
    for(int i = 0; i < 101; i++){  
        res[i] = '0';  
    }  
    for (int i = 100; i >= 0; i--){  
        char tmp[101];  
        int k;  
        for (k = 0; k < i; k++){  
            tmp[100-k] = '0';  
        }  
  
        int c = 0, sum = 0;  
        for (int j = 100; j >= 0; j--){  
            sum = ((int)num1[i] - 48) * ((int)num2[j] - 48) + c;  
            res[i+j] = sum % 10 + '0';  
            c = sum / 10;  
        }  
    }  
}
```

```
tmp[k] = (sum % 10) + 48;  
c = sum / 10;  
k--;  
if(k < 0) break;  
}  
  
add(res,tmp,res);  
}  
}  
  
// kiểm tra xem số nguyên lớn num1 có lớn hơn num2 không  
// Le Dinh Tri Tue - 0909 - 732833  
bool check(char *num1, char *num2){  
    int foo1, foo2;  
    for (foo1 = 0; foo1 < 101; foo1++){  
        if (num1[foo1] != '0') break;  
    }  
  
    for (foo2 = 0; foo2 < 101; foo2++){  
        if (num2[foo2] != '0') break;  
    }  
  
    if (foo1 > foo2) return false;  
    else if (foo1 < foo2) return true;  
    else {  
        int foo = foo1;  
        while (foo < 101){  
            if (num1[foo] < num2[foo]) return false;  
            else if (num1[foo] > num2[foo]) return true;  
            else foo++;  
        }  
    }  
}
```

```
    }

}

return true;

}

// định nghĩa cho operator +
// Le Dinh Tri Tue - 0909 - 732833

bigNum operator + (bigNum num1, bigNum num2){

    bigNum res;

    if (num1.sign == '1' && num2.sign == '1'){

        res.sign = '1';

        add(res.num, num1.num, num2.num);

        return res;

    } else if (num1.sign == '1' && num2.sign == '0'){

        if (check(num1.num, num2.num)){

            res.sign = '1';

            sub(res.num, num1.num, num2.num);

            return res;

        } else {

            res.sign = '0';

            sub(res.num, num2.num, num1.num);

            return res;

        }

    } else if (num1.sign == '0' && num2.sign == '1'){

        if (check(num1.num, num2.num)){

            res.sign = '0';

            sub(res.num, num1.num, num2.num);

            return res;

        }

    }

}
```

```
    } else {
        res.sign = '1';
        sub(res.num, num2.num, num1.num);
        return res;
    }
} else {
    res.sign = '0';
    add(res.num, num1.num, num2.num);
    return res;
}
}

// định nghĩa cho operator -
// Le Dinh Tri Tue - 0909 - 732833

bigNum operator - (bigNum num1, bigNum num2){
    bigNum res;

    if (num1.sign == '1' && num2.sign == '0'){
        num2.sign = '1';
        res = num1 + num2;
        return res;
    } else if (num1.sign == '1' && num2.sign == '1'){
        num2.sign = '0';
        res = num1 + num2;
        return res;
    } else if (num1.sign == '0' && num2.sign == '1'){
        num2.sign = '0';
        res = num1 + num2;
        return res;
    } else {

```

```
    num2.sign = '1';
    res = num1 + num2;
    return res;
}

}

// định nghĩa cho operator *
// Le Dinh Tri Tue - 0909 - 732833
bigNum operator * (bigNum num1, bigNum num2){
    bigNum res;

    if (num1.sign == '1' && num2.sign == '1'){
        res.sign = '1';
        multi(res.num, num1.num, num2.num);
        return res;
    } else if (num1.sign == '1' && num2.sign == '0'){
        res.sign = '0';
        multi(res.num, num1.num, num2.num);
        return res;
    } else if (num1.sign == '0' && num2.sign == '1'){
        res.sign = '0';
        multi(res.num, num1.num, num2.num);
        return res;
    } else {
        res.sign = '1';
        multi(res.num, num1.num, num2.num);
        return res;
    }
}
```

```
// print BigNum  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
void printBigNumber(bigNum number){  
  
    cout << number.sign;  
  
    int start;  
  
    for (start = 0; start < 101; start++){  
  
        if (number.num[start] != '0') break;  
    }  
  
    for (int i = start; i < 101; i++){  
  
        cout << number.num[i];  
    }  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
  
    bigNum num1, num2;  
  
    input(num1,num2);  
  
  
    bigNum so3, so4;  
  
    so3.sign = '1', so4.sign = '1';  
  
    for (int i = 0; i < 100; i++){  
  
        so3.num[i] = '0';  
        so4.num[i] = '0';  
    }  
  
    so3.num[100] = 3 + 48;  
    so4.num[100] = 4 + 48;  
  
  
    bigNum res = num1*num2 - so3 * num1 + so4 * num2;  
  
  
    printBigNumber(res);  
}
```

// Le Dinh Tri Tue - 0909 – 732833

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
f (?) { g++ tesst.cpp -o tesst } ; if (?) { .\tesst }
0121807015

1347227347
042294724910108772
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
f (?) { g++ tesst.cpp -o tesst } ; if (?) { .\tesst }
0800547253714

0389013676936
1311423830729145609193702
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
f (?) { g++ tesst.cpp -o tesst } ; if (?) { .\tesst }
1562862701008461237669505996967049208066942997054417216894858422269

1995808870599753186889299576925652087713749800604075758946132
12421824521203240690959994610073539746266906532596685834222631697937399112080780010140762062140731229
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT"
f (?) { g++ tesst.cpp -o tesst } ; if (?) { .\tesst }
1233565750636158306244820735298257795957130202802668419171402333

06461365948045416069564099230023646173885469440616221232595
063232700453674335664686358003691913808521484082405989034863646555726796680527239783370961938317781514
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH02-732833-Tuan12-TueLDT> █
```

Figure 66

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 3
MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ

MSSV: 20210909

Mã lớp: 732833

Contents

Bài thực hành số 3 – Tuần 14.....	80
Bài tập 3.1. Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ với $L_0 = 2, L_1 = 1$. Hãy viết hàm tính số Lucas thứ n.....	80
Bài tập 3.2. Trên bàn cờ vua kích thước $n*n$ có một quân mã đang ở ô $(1, 1)$. Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần ($\hat{o}(1, 1)$ được xem là đã đi qua).	81
Bài tập 3.3. Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là c_{ij} . Hãy tìm tổng chi phí nhỏ nhất có thể.....	84
Bài tập 3.4. Cho dãy a có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi một số phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a.	87
Bài tập 3.5. Tính hệ số tổ hợp $C(n, k)$	90
Bài tập 3.6. Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.	92
Bài tập 3.7. Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp.....	93
Bài tập 3.8. Cân đĩa: Bạn đang muốn kiểm tra xem một vật cho trước có đúng nặng M như người ta nói hay không. Có một cân thăng bằng và n quả cân. Quả thứ i nặng mi. Hãy chỉ ra một cách cân thỏa mãn. Quy cách in ra đã được tích hợp trong mã nguồn dưới.....	95
Bài tập 3.9. Lập lịch cho y tá: Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đa làm việc liên tiếp đều có số ngày thuộc đoạn $[K_1, K_2]$. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển.....	98
Bài tập 3.10. Khoảng cách Hamming giữa hai xâu cùng độ dài là số vị trí mà ký tự tại vị trí đó là khác nhau trên hai xâu. Cho S là xâu gồm n ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài n, có khoảng cách Hamming với S bằng H. Các xâu phải được liệt kê theo thứ tự từ điển.....	102
Bài tập 3.11. Lịch trình chụp ảnh Superior là một hòn đảo tuyệt đẹp với n địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau. Đoàn khách tham quan có r người với sở thích chụp ảnh khác nhau. Theo đó, mỗi người sẽ đưa ra danh sách các địa điểm mà họ muốn chụp. Bạn cần giúp mỗi người trong đoàn lập lịch di chuyển sao cho đi qua các điểm họ yêu cầu đúng một lần, không đi qua điểm nào khác, bắt đầu tại điểm đầu tiên và kết thúc tại điểm cuối cùng trong danh sách mà họ đưa ra, và có tổng khoảng cách đi lại là nhỏ nhất.....	106
Bài tập 3.12. Đếm đường đi Cho đồ thị vô hướng G, hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần.....	111

Figure 1	80
Figure 2	80
Figure 3	81
Figure 4	81
Figure 5	82
Figure 6	83
Figure 7	84
Figure 8	84
Figure 9	85
Figure 10	86
Figure 11	87
Figure 12	87
Figure 13	88
Figure 14	89
Figure 15	90
Figure 16	90
Figure 17	91
Figure 18	92
Figure 19	93
Figure 20	93
Figure 21	94
Figure 22	95
Figure 23	101
Figure 24	102
Figure 25	105
Figure 26	106
Figure 27	110
Figure 28	110
Figure 29	113
Figure 30	114
Figure 31	115
Figure 32	116

Bài thực hành số 3 – Tuần 14

Copy code (kèm ảnh code và kết quả vào)

Bài tập 3.1. Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ với $L_0 = 2$, $L_1 = 1$. Hãy viết hàm tính số Lucas thứ n.

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

```

1 /*Bài 3.1. Dãy Lucas được định nghĩa bởi
2 Ln = Ln-1 + Ln-2 với L0 = 2, L1 = 1. Hãy viết hàm tính số Lucas thứ n.
3 */
4 int lucas(int n) {
5
6     //*****
7     // Le Dinh Tri Tue - 0909 - 732833
8     if(n == 0) return 2;
9     if(n == 1) return 1;
10    return lucas(n-1) + lucas(n-2);
11    //*****
12 }
```

Figure 67

```

20
21 // Le Dinh Tri Tue - 0909 - 732833

```

Precheck
Check

Test	Expected	Got	
✓ cout << lucas(5);	11	11	✓
✓ cout << lucas(10);	123	123	✓
✓ cout << lucas(30);	1860498	1860498	✓

Passed all tests! ✓

Figure 68

Bài tập 3.2. Trên bàn cờ vua kích thước $n \times n$ có một quân mã đang ở ô $(1, 1)$. Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần ($\text{ô } (1, 1)$ được xem là đã đi qua).

```

1  /*Bài 3.2. Trên bàn cờ vua kích thước n*n có một
2  quân mã đang ở ô (1, 1). Hãy đưa ra một dãy các
3  di chuyển của mã sao cho mỗi ô trên bàn cờ đều được
4  đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua).
5 */
6 #include<stdio.h>
7 #include<stdlib.h>
8 #include <iostream>
9 using namespace std;
10 // Le Dinh Tri Tue - 0909 - 732833
11 int n;
12 int X[100], Y[100]; //# Lưu tọa độ các bước di chuyển của quân mã
13 int mark[100][100]; //# Đánh dấu vị trí các ô mà quân mã đã di chuyển qua
14
15 //# Mảng hx, hy mô tả 8 vị trí quân mã có thể di chuyển kể từ vị trí hiện tại
16 const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
17 const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};
18 // Le Dinh Tri Tue - 0909 - 732833
19 //# In ra dãy các di chuyển tìm được
20 void print_sol(){
21     for (int j = 1; j <= n * n; ++j)
22         printf("(%d %d)\n", X[j], Y[j]);

```

Figure 69

```

21     for (int j = 1; j <= n * n; ++j)
22         printf("(%d %d)\n", X[j], Y[j]);
23     exit(0);
24 }
25 // Le Dinh Tri Tue - 0909 - 732833
26 bool check(int xx, int yy){
27     if(mark[xx][yy] == 1) return false;
28     if(xx < 1 || xx > n || yy < 1 || yy > n) return false;
29     //cout << xx << " " << yy << endl;
30     return true;
31 }
32
33 //# Thuật toán quay lui
34 // Le Dinh Tri Tue - 0909 - 732833
35 void TRY(int k){
36     for(int i = 0; i < 8; i++){
37         int xx = X[k-1] + hx[i];
38         int yy = Y[k-1] + hy[i];
39         // Le Dinh Tri Tue - 0909 - 732833
40         if(check(xx,yy)){
41             mark[xx][yy] = 1;
42             X[k] = xx;

```

Figure 70

```
40 |     if(check(xx,yy)){
41 |         mark[xx][yy] = 1;
42 |         X[k] = xx;
43 |         Y[k] = yy;
44 |
45 |         if(k == n*n) print_sol();
46 |         else TRY(k+1);
47 |
48 |         mark[xx][yy] = 0;
49 |     }
50 |     *****/
51 |
52 |
53 |
54 int main(){
55 // Le Dinh Tri Tue - 0909 - 732833
56 cin >> n;
57 mark[1][1] = 1;
58 X[1] = Y[1] = 1;
59 TRY(2);
60 return 0;
61 }
```

Figure 71

```
54 v int main(){
55     // Le Dinh Tri Tue - 0909 - 732833
56     cin >> n;
57     mark[1][1] = 1;
58     X[1] = Y[1] = 1;
59     TRY(2);
60     return 0;
61 }
```

Precheck

Check

	Input	Expected	Got		
✓	5	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4) (2 2) (4 1) (5 3) (4 5) (2 4) (1 2) (3 1) (5 2) (4 4) (2 5) (1 3)	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4) (2 2) (4 1) (5 3) (4 5) (2 4) (1 2) (3 1) (5 2) (4 4) (2 5) (1 3)	✓	

Figure 72

Bài tập 3.3. Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể.

```

1 /*
2 Bài 3.3. Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các
3 thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi
4 từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể
5 */
6 #include <bits/stdc++.h>
7 using namespace std;
8 #define MAX 100
9 // Le Dinh Tri Tue - 0909 - 732833
10 int n, c[MAX][MAX]; //## số thành phố và ma trận chi phí
11 int cmin = INT_MAX; //## chi phí di lại nhõ nhất giữa hai thành phố khác nhau
12 int best = INT_MAX; //## tổng chi phí nhõ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT_MAX = 2^31-1
13 int curr; //## tổng chi phí tới thời điểm hiện tại
14 int mark[MAX]; //## đánh dấu những thành phố đã đi
15 int x[MAX]; //## lưu giữ các thành phố đã đi
16
17 ## Đọc dữ liệu vào
18 // Le Dinh Tri Tue - 0909 - 732833
19 void input(){
20     cin >> n;
21     for (int i = 1; i <= n; ++i){
22         for (int j = 1; j <= n; ++j){
23             cin >> c[i][j];
24             if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
25         }
26         for(int i=0; i<MAX; i++)
27             mark[i] = 0;
28         curr = 0;
29     }
30
31 bool check(int k, int i){
32     if(mark[i] == 1) return false;
33     return true;
34 }
35 // Le Dinh Tri Tue - 0909 - 732833
36 void solution(){
37     best = min(best,curr);
38 }
39
40 ## Thuật toán quay lui

```

Figure 73

```

19 void input(){
20     cin >> n;
21     for (int i = 1; i <= n; ++i)
22         for (int j = 1; j <= n; ++j){
23             cin >> c[i][j];
24             if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
25         }
26         for(int i=0; i<MAX; i++)
27             mark[i] = 0;
28         curr = 0;
29     }
30
31 bool check(int k, int i){
32     if(mark[i] == 1) return false;
33     return true;
34 }
35 // Le Dinh Tri Tue - 0909 - 732833
36 void solution(){
37     best = min(best,curr);
38 }
39
40 ## Thuật toán quay lui

```

Figure 74

```
40 //# Thuật toán quay lui
41 // Le Dinh Tri Tue - 0909 - 732833
42 void TRY(int k){
43     for(int i = 2; i <= n; i++){
44         // Le Dinh Tri Tue - 0909 - 732833
45         if(check(k,i)){
46             mark[i] = 1;
47             curr += c[x[k-1]][i];
48
49             x[k] = i;
50
51             if(k == n) {
52                 curr += c[x[n]][1];
53                 solution();
54                 curr -= c[x[n]][1];
55             }
56             else TRY(k+1);
57
58             mark[i] = 0;
59             curr -= c[x[k-1]][i];
60         }
61     }  
*****
```

Figure 75

```
54     curr -= c[x[n]][1];
55 }
56 else TRY(k+1);
57
58 mark[i] = 0;
59 curr -= c[x[k-1]][i];
60 }
61 *****/
62 }
63 }
64
65 int main() {
66 // Le Dinh Tri Tue - 0909 - 732833
67 input();
68 x[1] = 1;
69 TRY(2);
70 cout << best;
71 return 0;
72 }
```

Precheck

Check

	Input	Expected	Got	
✓	4 0 2 1 3 4 0 1 2 2 1 0 3 3 4 2 0	7	7	✓
✓	6 0 2 1 3 7 3 4 0 1 2 8 5 2 1 0 3 6 9 3 4 2 0 2 3 1 7 3 9 0 4	11	11	✓

Figure 76

Bài tập 3.4. Cho dãy a có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi một số phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a.

```

1 /*
2 Bài 3.4. Cho dãy a có n phần tử. Một dãy con của a là
3 dãy thu được bằng cách xóa đi một số phần tử của a và
4 giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào).
5 Hãy tìm dãy con tăng dài nhất của a.
6 */
7 #include <bits/stdc++.h>
8 using namespace std;
9 int a[1000], n;
10 int mem[1000]; //# mảng ghi nhớ lời giải các bài toán con đã được giải
11 // Le Dinh Tri Tue - 0909 - 732833
12 void init(){
13     memset(mem, -1, sizeof(mem));
14 }
15
16 //# Quy hoạch động,
17 //# Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
18
19 // Le Dinh Tri Tue - 0909 - 732833
20 void incsubseq(){
21     mem[0] = 1;
22 }
```

Figure 77

```

22
23 for(int i=1; i<n; i++){
24     mem[i] = 1;
25     for(int j=0; j<i; j++){
26         if(a[j] < a[i]){
27             if(mem[i] < mem[j] + 1){
28                 mem[i] = mem[j] + 1;
29             }
30         }
31     }
32 }
33 }
34 int lis(int i) {
35     // Le Dinh Tri Tue - 0909 - 732833
36     return mem[i];
37     ****
38 }
39
40 // Truy vết lời giải
41 // Le Dinh Tri Tue - 0909 - 732833
42 void trace(int i){
43     for(int j = 0; j < i; j++){
        ...
```

Figure 78

```
40 // nhanh vèo với gõ đúp
41 // Le Dinh Tri Tue - 0909 - 732833
42 void trace(int i){
43     for(int j = 0; j < i; j++){
44         if (a[j] < a[i] && mem[i] == 1 + mem[j]){
45             trace(j);
46             break;
47         }
48     }
49     cout << a[i] << " ";
50 }
51
52 int main(){
53     // Le Dinh Tri Tue - 0909 - 732833
54     init();
55     cin >> n;
56     for(int i = 0; i < n; i++) cin >> a[i];
57
58     incsubseq();
59
60     int res = 1, pos = 0;
61     for(int i = 1; i < n; i++){
62         :r -----
```

Figure 79

```
54     int n;
55     cin >> n;
56     for(int i = 0; i < n; i++) cin >> a[i];
57
58     incsubseq();
59
60     int res = 1, pos = 0;
61     for(int i = 1; i < n; i++){
62         if (res < lis(i)){
63             res = lis(i);
64             pos = i;
65         }
66     }
67     cout << res << endl;
68     trace(pos);
69     return 0;
70 }
```

Precheck

Check

	Input	Expected	Got	
✓	6 2 1 5 4 3 6	3 2 5 6	3 2 5 6	✓
✓	10 2 1 5 4 3 6 3 -6 9 10	5 2 5 6 9 10	5 2 5 6 9 10	✓
✓	20 2 -10 -8 5 4 3 6 3 -6 9 10 12 2 5 25 9 15 30 -100 45	10 -10 -8 5 6 9 10 12 25 30 45	10 -10 -8 5 6 9 10 12 25 30 45	✓

Passed all tests! ✓

Figure 80

Bài tập 3.5. Tính hệ số tổ hợp C(n, k).

```

1 //Bài 3.5. Tính hệ số tổ hợp C(n, k)
2 // Le Dinh Tri Tue - 0909 - 732833
3 #include<stdio.h>
4 #include <iostream>
5 using namespace std;
6 const int MAX = 1000;
7 // Le Dinh Tri Tue - 0909 - 732833
8 int binom(int n, int k) {
9     if (k > n) return 0;
10    if (k == 0) return 1;
11    return binom(n-1, k) + binom(n-1, k-1);
12 }
13 // Le Dinh Tri Tue - 0909 - 732833
14 int binom2(int n, int k){
15
16     //# Khử đệ quy
17     // Le Dinh Tri Tue - 0909 - 732833
18     long result = 1; //(nC0 = 1)
19     for ( int i = 1; i <= k ; i++)
20     {
21         result = result * n / i;
22         n--;

```

Figure 81

```

27
28 int main() {
29     // Le Dinh Tri Tue - 0909 - 732833
30     int m;
31     cin >> m;
32     for (int n = 1; n <= m; ++n){
33         for (int k = 0; k <= n; ++k)
34             printf("%d ", binom(n, k));
35             printf("\n");
36     }
37     for (int n = 1; n <= m; ++n){
38         for (int k = 0; k <= n; ++k)
39             printf("%d ", binom2(n, k));
40             printf("\n");
41     }
42     return 0;
43 }

```

Figure 82

```
27  
28 int main() {  
29     // Lê Đình Trí Tue - 0909 - 732833  
30     int m;  
31     cin >> m;  
32     for (int n = 1; n <= m; ++n){  
33         for (int k = 0; k <= n; ++k)  
34             printf("%d ", binom(n, k));  
35         printf("\n");  
36     }  
37     for (int n = 1; n <= m; ++n){  
38         for (int k = 0; k <= n; ++k)  
39             printf("%d ", binom2(n, k));  
40         printf("\n");  
41     }  
42     return 0;  
43 }
```

Precheck

Check

	Input	Expected
✓	4	1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
✓	10	1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1

Figure 83

Bài tập 3.6. Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.

```
1 /*Bài 3.6. Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.
2 */
3 #include <iostream>
4 using namespace std;
5 // Le Dinh Tri Tue - 0909 - 732833
6 int gcd(int a, int b){
7     if (b == 0) return a;
8     return gcd(b, a % b);
9 }
10 // Le Dinh Tri Tue - 0909 - 732833
11 int gcd2(int a, int b){
12
13     //# Khử đệ quy
14     // Le Dinh Tri Tue - 0909 - 732833
15     while(b != 0){
16         int tmp = a % b;
17         a = b;
18         b = tmp;
19     }
20
21     return a;
22 //******/
```

Figure 84

```

23 |
24 | int main() {
25 v   // Le Dinh Tri Tue - 0909 - 732833
26   int a, b;
27   cin >> a >> b;
28   cout << gcd(a, b) << endl << gcd2(a, b);
29   return 0;
30 }
31 }
```

Precheck Check

	Input	Expected	Got	
✓	50 35	5 5	5 5	✓
✓	217 413	7 7	7 7	✓

Passed all tests! ✓

Figure 85

Bài tập 3.7. Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp.

```

1 v /*Bài 3.7. Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê
2 các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp
3 */
4 #include <bits/stdc++.h>
5 using namespace std;
6 // Le Dinh Tri Tue - 0909 - 732833
7 v struct state{
8   int i, j, old_L;
9   // constructor
10  state(int _i = 0, int _j = 0, int _L = 0):
11    i(_i), j(_j), old_L(_L){}
12};
13
14 v int main() {
15   // Le Dinh Tri Tue - 0909 - 732833
16   int n, k;
17   cin >> n >> k;
18   int x[n+1];
19   stack<state> s;
20   //## number of consecutive suffix 1
21   int L = 0;
22   s.push(state(1, 0));
```

Figure 86

```
19      stack<state> s,
20      //# number of consecutive suffix 1
21      int L = 0;
22      s.push(state(1, 0));
23  ▾ while (!s.empty()){
24      state &top = s.top();
25      //# if a new binary sequence is found
26  ▾ if (top.i > n){
27          for (int i = 1; i <= n; ++i)
28              cout << x[i] << " \n"[i == n];
29          s.pop();
30          continue;
31      }
32
33      //# Khử đệ quy
34      // Le Dinh Tri Tue - 0909 - 732833
35
36      if(top.j>0) L = top.old_L;
37
38  ▾ if(top.j>1){
39          s.pop();
40          continue;
41      }
```

Figure 87

```

35
36     if(top.j>0) L = top.old_L;
37
38     if(top.j>1){
39         s.pop();
40         continue;
41     }
42
43     if(L+1<k || top.j==0){
44         x[top.i] = top.j;
45         top.old_L = L;
46         if(top.j){
47             L = L + 1;
48         } else {L = 0;}
49         s.push(state(top.i+1,0));
50     }
51     top.j++;
52 // Le Dinh Tri Tue - 0909 - 732833
53 ****
54 }
55 return 0;
56 }
```

Precheck

Check

	Input	Expected	Got	
✓	4 2	0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0	0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0	✓

Figure 88

Bài tập 3.8. Cân đĩa: Bạn đang muốn kiểm tra xem một vật cho trước có đúng nặng M như người ta nói hay không. Có một cân thăng bằng và n quả cân. Quả thứ i nặng mi. Hãy chỉ ra một cách cân thỏa mãn. Quy cách in ra đã được tích hợp trong mã nguồn dưới.

CODE:

/*

Bài tập 8: Cân đĩa

Bạn đang muốn kiểm tra xem một vật cho trước có đúng nặng M

như người ta nói hay không. Có một cân thăng bằng và n

quả cân. Quả thứ i nặng mi. Hãy chỉ ra một cách cân thỏa mãn.

Quy cách in ra đã được tích hợp trong mã nguồn dưới.*/

```
#include <bits/stdc++.h>

using namespace std;

// Le Dinh Tri Tue - 0909 - 732833

struct state{

    int i, j, s;

    state(int _i = 0, int _j = 0, int _s=0): i(_i), j(_j), s(_s){}

};

// Le Dinh Tri Tue - 0909 - 732833

int main() {

    int n, M;

    cin >> n >> M;

    int m[n+1];

    for (int i = 1; i <= n; ++i) cin >> m[i];

    int x[n+1];

    stack<state> s;

    //# sum of selected weights

    // Le Dinh Tri Tue - 0909 - 732833

    int sum = 0;

    s.push(state(1, -1, -m[1]));

    s.push(state(1, 0, 0));

    s.push(state(1, 1, m[1]));

    while (!s.empty()){

        state top = s.top();

        if (top.i >= n){

            if (top.s == M){

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

                    if (x[i] == -1) cout << '-' << m[i];

                    if (x[i] == 1) cout << '+' << m[i];

```

```
        }

        cout << "=" << M;

        exit(0);

    }

    s.pop();

    continue;

}

// Le Dinh Tri Tue - 0909 - 732833

//# Khử đê quy

s.pop();

x[top.i]=top.j;

s.push(state(top.i+1, -1, top.s-m[top.i+1]));

s.push(state(top.i+1, 0,top.s));

s.push(state(top.i+1, 1,top.s+m[top.i+1]));

}

cout << -1;

return 0;
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Use
ile } ; if ($?) { .\tempCodeRunnerFile }
6 10
7 1 2 3 4 5
+7+1+2=10
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> █
```

Bài tập 3.9. Lập lịch cho y tá: Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đa làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển.

CODE:

```
/*
```

Bài tập 9: Lập lịch cho y tá

Một y tá cần lập lịch làm việc trong N ngày,

mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi.

Một lịch làm việc là tốt nếu không có hai ngày

nghỉ nào liên tiếp và mọi chuỗi ngày tối đa

làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2].

Hãy liệt kê tất cả các cách lập lịch tốt,

với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n

với bit 0/1 tương ứng là nghỉ/làm việc.

Các xâu phải được in ra theo thứ tự từ điển*/

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
const int MAX = 1000;
```

```
const int du = 1000000000 + 7;
```

```
int n, k1, k2;
```

```
int x[MAX];
```

```
int cnt = 0;
```

```
int so0 = 0, so1 = 0;
```

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
void inputData(){
```

```
    cin >> n >> k1 >> k2;
```

```
}
```

```
bool check(int a, int i){  
    if(a==1) return true;  
    // Le Dinh Tri Tue - 0909 - 732833  
    else {  
        // neu ngay nay nghi -> kiem tra ngay truoc do, neu da nghi thi khong duoc nghi nua  
        if(i==0){  
            if(x[a-1] == 0) return false;  
            else { // x[a-1] == 1  
                if(so1<k1) return false;  
            }  
        } else { // i == 1  
            if(x[a-1] == 0){  
                if(n-a+1 < k1) return false;  
            } else {  
                if(so1>=k2) return false;  
            }  
        }  
        // Le Dinh Tri Tue - 0909 - 732833  
        return true;  
    }  
  
}  
  
void solution(){  
    // Le Dinh Tri Tue - 0909 - 732833  
    for(int i=1; i<=n; i++)  
        cout << x[i];  
    cout << endl;  
    cnt++;  
}
```

```
// Le Dinh Tri Tue - 0909 - 732833

void TRY(int a){

    for(int i=0; i<=1; i++){
        if(check(a,i)){
            x[a] = i; // ngay thu a lam viec hoac khong lam viec
            int pre = so1;
            if(i == 1){
                if(x[a-1] == 1) so1++;
                else so1 = 1;
            } else {
                so1 = 0;
            }

            if(a==n) solution();
            else TRY(a+1);

            so1 = pre;
        }
    }
}

// Le Dinh Tri Tue - 0909 - 732833

int main(){

    inputData();
    TRY(1);
    return 0;
}
```

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\unnerFile" ; if ($?) { .\tempCodeRunnerFile }
6 2 3
011011
110110
110111
111011
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\unnerFile" ; if ($?) { .\tempCodeRunnerFile }
10 2 3
0110110110
0110110111
0110111011
0111011011
1101101110
1101110110
1101110111
1110110110
1110110111
1110111011
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\unnerFile" ; if ($?) { .\tempCodeRunnerFile }
20 4 7
011110111101111011110
01111011111011111110
01111011111101111110
01111011111101111111
01111011111101111110
01111011111101111111
011111011111101111110
011111011111101111111
011111011111101111110
011111011111101111111
011111011111101111110
0111110111111101111110
0111110111111101111111
0111111011111101111110
01111110111111101111111
01111110111111101111110
01111110111111101111111
01111110111111101111110
01111111011111101111111
```

Figure 89

PS D:\Users\ADMIN\Documents\1.Thuc Hanh Ki Thua Lap Trinh>

Figure 90

Bài tập 3.10. Khoảng cách Hamming giữa hai xâu cùng độ dài là số vị trí mà ký tự tại vị trí đó là khác nhau trên hai xâu. Cho S là xâu gồm n ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài n , có khoảng cách Hamming với S bằng H . Các xâu phải được liệt kê theo thứ tự từ điển.

CODE:

/*

Bài tập 10: Khoảng cách Hamming giữa hai xâu cùng độ dài là số vị trí mà

ký tự tại vị trí đó là khác nhau trên hai xâu. Cho S là xâu gồm n

ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài n, có

khoảng cách Hamming với S bằng H . Các xâu phải được liệt kê theo thứ tự từ điển*/

```
#include<bits/stdc++.h>

using namespace std;

const int MAX = 20;

int N, H;

int x[MAX];

int S[MAX]; // Xau mac dinh de so sanh

int cnt = 0;

// Le Dinh Tri Tue - 0909 - 732833

void input(){

    cin >> N >> H;

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

        S[i] = 0;

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

            x[i] = 0;

        }

    }

    // Le Dinh Tri Tue - 0909 - 732833

    // Kiem tra 2 co khoang cach Hamming la bao nhieu

    int checkHamming(int str1[], int str2[]){

        int lens = N;

        int cnt = 0;

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

            if(str1[i] != str2[i]) cnt++;

        }

        return cnt;

    }

    // Le Dinh Tri Tue - 0909 - 732833

    bool check(int a, int i){

        return true;

    }

    // Le Dinh Tri Tue - 0909 - 732833
```

```
void solution(){
    if(checkHamming(x,S) == H){
        for(int i=0; i<N; i++)
            cout << x[i];
        cout << endl;
    }
}

// Le Dinh Tri Tue - 0909 - 732833

void TRY(int a){
    for(int i=0; i<=1; i++){
        x[a] = i;
        if(a == N-1) solution();
        else
            TRY(a+1);
    }
}

// Le Dinh Tri Tue - 0909 - 732833

int main(){
    int T;
    cin >> T;
    while(T > 0){
        input();
        TRY(0);
        T--;
    }
}
```

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "dPS D:\Users\ADMIN\_
_Lap_Trinh\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if (0
7

4 2
0011
0101
0110
1001
1010
1100

1 0
0

1 1
1

2 0
00

2 1
01
10

5 2
00011
00101
00110
01001
01010
01100
10001
10010
10100
11000
```

Figure 91

```

2 1
01
10

5 2
00011
00101
00110
01001
01010
01100
10001
10010
10100
11000

5 3
00111
01011
01101
01110
10011
10101
10110
11001
11010
11100
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> █

```

Figure 92

Bài tập 3.11. Lịch trình chụp ảnh Superior là một hòn đảo tuyệt đẹp với n địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau. Đoàn khách tham quan có r người với sở thích chụp ảnh khác nhau. Theo đó, mỗi người sẽ đưa ra danh sách các địa điểm mà họ muốn chụp. Bạn cần giúp mỗi người trong đoàn lập lịch di chuyển sao cho đi qua các điểm họ yêu cầu đúng một lần, không đi qua điểm nào khác, bắt đầu tại điểm đầu tiên và kết thúc tại điểm cuối cùng trong danh sách mà họ đưa ra, và có tổng khoảng cách đi lại là nhỏ nhất.

CODE:

```

/*Bài tập 11: Lịch trình chụp ảnh Superior là một hòn đảo tuyệt đẹp với n
địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau.

Đoàn khách tham quan có r người với sở thích chụp ảnh khác nhau.

Theo đó, mỗi người sẽ đưa ra danh sách các địa điểm mà họ muốn chụp.

Bạn cần giúp mỗi người trong đoàn lập lịch di chuyển sao cho đi qua các điểm
họ yêu cầu đúng một lần, không đi qua điểm nào khác, bắt đầu tại điểm đầu tiên
và kết thúc tại điểm cuối cùng trong danh sách mà họ đưa ra, và có tổng khoảng cách đi lại là nhỏ
nhất.*/

```

// Le Dinh Tri Tue - 0909 - 732833

```
#include<bits/stdc++.h>

using namespace std;

const int MAX = 10000;

int n, r;

int price[MAX][MAX];

int x[MAX];

bool visited[MAX];

vector<int> vt;

int min_price;

int sum_price;

int start, destination, numberPoint;

// Le Dinh Tri Tue - 0909 - 732833

void input(){

    cin >> n >> r;

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

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

            cin >> price[i][j];

        }

    }

    // Le Dinh Tri Tue - 0909 - 732833

    bool check(int a, int i){

        if(visited[vt[i]]) return false;

        if(price[x[a-1]][vt[i]] == 0) return false;

        return true;

    }

    // Le Dinh Tri Tue - 0909 - 732833

    void solution(){

        if(price[x[numberPoint-2]][destination] == 0) return;

        min_price = min(min_price, sum_price + price[x[numberPoint-2]][destination]);

    }

}
```

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
void TRY(int a){  
    for(int i=1; i<numberOfPoint-1; i++){  
        if(check(a, i)){  
            visited[vt[i]] = true;  
            sum_price += price[x[a-1]][vt[i]];  
  
            x[a] = vt[i];  
            if(a == numberOfPoint-2) solution();  
            else TRY(a+1);  
  
            visited[vt[i]] = false;  
            sum_price -= price[x[a-1]][vt[i]];  
        }  
    }  
}
```

```
// Le Dinh Tri Tue - 0909 - 732833
```

```
int main(){  
    string str;  
    input(); getline(cin,str);  
  
    while(r > 0){  
        min_price = INT_MAX;  
        sum_price = 0;  
        // Le Dinh Tri Tue - 0909 - 732833  
        getline(cin, str);  
        while (!str.empty()){  
            stringstream convert(str.substr(0, str.find(" ")));  
            int tmp = 0;  
            convert >> tmp;
```

```
vt.push_back(tmp - 1);

if (str.find(" ") > str.size()){

    break;

} else {

    str.erase(0, str.find(" ") + 1); // Update string

}

}

// Le Dinh Tri Tue - 0909 - 732833

// Bat dau khoi tao cac du lieu can thiet truoc khi quay lui

start = vt[0]; // diem bat dau dau

destination = vt[vt.size()-1]; // diem dich

numberOfPoint = vt.size(); // so diem phai di qua

x[0] = start; x[numberOfPoint-1] = destination;

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

    visited[i] = false;

TRY(1);

// Le Dinh Tri Tue - 0909 - 732833

// In ra ket qua

if(min_price == INT_MAX) cout << "0" << endl;

else cout << min_price << endl;

// Xoa vector va chuyen sang khach tiep theo

vt.erase(vt.begin(), vt.end());

r--;

}

}
```

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
unnerFile } ; if ($?) { .\tempCodeRunnerFile }

6 3
0 1 2 0 1 1
1 0 1 1 1 0
0 2 0 1 3 0
4 3 1 0 0 0
0 0 1 1 0 0
1 0 0 0 0 0
6 3 2 5
0
6 1 2 3 4 5
7
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
unnerFile } ; if ($?) { .\tempCodeRunnerFile }

20 5
8 9 9 1 7 5 5 10 1 0 7 7 5 8 6 7 3 7 9 2
7 7 8 10 6 7 8 5 6 7 2 1 9 10 10 2 4 9 7 5
4 7 5 6 7 0 5 0 5 4 5 1 3 10 6 0 0 6 7 8
7 10 3 5 2 10 9 10 3 10 1 1 6 8 5 5 2 4 6 6
5 6 5 0 1 4 5 6 5 7 4 0 8 4 4 0 3 3 5 8
4 7 8 2 0 8 7 7 4 1 5 2 7 9 0 10 3 8 4 1
6 2 9 2 3 6 5 9 5 7 4 6 1 10 7 9 10 7 0 7
2 0 0 10 7 10 1 1 10 9 9 9 6 6 1 7 9 9 7 5
8 1 3 0 10 4 7 4 5 7 6 3 2 9 9 8 2 0 0 9
9 4 9 8 7 9 3 4 10 5 10 10 8 10 9 1 10 6 10 10

```

Figure 93

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
unnerFile } ; if ($?) { .\tempCodeRunnerFile }

20 5
8 9 9 1 7 5 5 10 1 0 7 7 5 8 6 7 3 7 9 2
7 7 8 10 6 7 8 5 6 7 2 1 9 10 10 2 4 9 7 5
4 7 5 6 7 0 5 0 5 4 5 1 3 10 6 0 0 6 7 8
7 10 3 5 2 10 9 10 3 10 1 1 6 8 5 5 2 4 6 6
5 6 5 0 1 4 5 6 5 7 4 0 8 4 4 0 3 3 5 8
4 7 8 2 0 8 7 7 4 1 5 2 7 9 0 10 3 8 4 1
6 2 9 2 3 6 5 9 5 7 4 6 1 10 7 9 10 7 0 7
2 0 0 10 7 10 1 1 10 9 9 9 6 6 1 7 9 9 7 5
8 1 3 0 10 4 7 4 5 7 6 3 2 9 9 8 2 0 0 9
9 4 9 8 7 9 3 4 10 5 10 10 8 10 9 1 10 6 10 10
9 4 4 5 8 2 8 8 6 9 9 8 3 5 7 10 5 3 2 10
9 3 10 10 1 10 5 7 10 7 3 2 9 8 3 9 1 5 3 5
10 1 7 1 5 9 4 8 5 5 10 5 6 2 0 8 6 2 10 9
4 1 7 8 2 10 4 6 0 1 4 8 7 2 4 9 4 2 4 7
8 3 3 10 9 0 8 3 4 0 6 10 6 8 9 0 0 9 3 7
0 1 3 4 8 6 2 5 5 8 5 9 4 2 2 4 10 3 0 0
10 2 0 2 6 7 8 0 5 1 2 2 3 5 8 8 10 2 0 9
1 1 0 0 10 6 6 1 1 3 7 0 7 9 8 8 8 4 1 5
2 1 8 5 10 0 2 6 5 4 7 8 4 2 8 8 8 5 1 4
10 5 6 10 5 6 0 3 8 7 8 2 7 7 8 9 7 5 5 4
14 11 4 9
12
13 6 3 1 14
25
8 3 6 4 5 13
19
2 4 13 6 14 12 3
27
3 7 5 1 8 14 15 11
25
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> █

```

Figure 94

Bài tập 3.12. Đếm đường đi Cho đồ thị vô hướng G, hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần.

CODE:

```
/*Bài tập 12: Đếm đường đi Cho đồ thị vô hướng G, hãy đếm số đường đi đi qua k
cạnh và không đi qua đỉnh nào quá một lần.*/
#include<bits/stdc++.h>
using namespace std;
const int MAX = 100;
int n, k;
int m;
vector<vector<int>> vt; // chuyen tu luu tru cac canh sang luu tru danh sach ke
int x[MAX];
bool visited[MAX];
int res;
// Le Dinh Tri Tue - 0909 - 732833

void input(){
    cin >> n >> k;
    cin >> m;
    vt.resize(n+1);
    for(int i=0; i<m; i++){
        int tmp1, tmp2;
        cin >> tmp1 >> tmp2;

        vt[tmp1-1].push_back(tmp2-1); // dinh tmp2 ke voi dinh tmp1
        vt[tmp2-1].push_back(tmp1-1); // dinh tmp1 ke voi dinh tmp2
    }
}

for(int i=0; i<n; i++){
    visited[i] = false;
}
```

```
}

res = 0;

}

// Le Dinh Tri Tue - 0909 - 732833

bool check(int a, int i){

    if(a == 0) return true;

    if(visited[i]) return false;

    int index = 0;

    for(int j=0; j<vt[x[a-1]].size(); j++){

        if(i == vt[x[a-1]][j]) index++;

    }

    if(index == 0) return false;

    return true;

}

// Le Dinh Tri Tue - 0909 - 732833

void solution(){

    res++;

}

void TRY(int a){

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

        if(check(a, i)){

            visited[i] = true;

            x[a] = i;

            if(a == k) solution();

            else TRY(a+1);

        }

    }

}
```

```
visited[i] = false;  
}  
}  
}  
  
// Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
    input();  
    TRY(0);  
    cout << res / 2;  
}
```

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "dPS D:\Users\ADMIN\Documents\1.Thu  
_Lap_Trinh\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if (?) { .\tempCo  
10 4  
20  
6 8  
1 2  
9 7  
5 9  
2 3  
7 4  
5 8  
1 8  
9 6  
3 5  
3 9  
4 2  
2 9  
1 7  
6 3  
10 3  
2 10  
6 10  
7 8  
3 1  
468  
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_H
```

Figure 95

```
unnerFile } ; if ($?) { .\tempCodeRunnerFile }
20 5
50
6 10
3 16
9 8
18 19
20 18
14 5
19 13
12 17
14 18
11 20
13 15
8 12
2 19
5 17
16 8
18 11
1 15
12 1
16 13
8 5
7 8
4 18
14 6
13 4
15 14
1 3
17 15
14 13
4 15
20 14
20 17
8 13
14 2
14 8
9 6
20 5
4 8
```

Figure 96

```
9 6
20 5
4 8
3 7
1 20
4 11
1 8
14 10
16 10
12 6
11 2
8 6
14 9
15 6
16 14
16 20
16825
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
unnerFile } ; if ($?) { .\tempCodeRunnerFile }
20 5
50
18 3
8 7
8 2
15 7
12 3
18 5
15 3
10 6
17 13
6 18
3 19
6 17
19 8
9 6
10 2
8 9
15 19
1 10
1 16
```

Figure 97

```
10 2
8 9
15 19
1 10
1 16
1 15
4 1
12 10
14 10
5 20
15 17
4 14
1 3
2 1
14 15
1 13
4 15
8 12
14 18
19 16
19 6
16 11
1 8
4 5
1 12
4 2
13 11
14 11
7 10
2 14
1 18
10 17
13 4
3 4
11 6
9 19
16565
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> █
```

Figure 98

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 4
MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ

MSSV: 20210909

Mã lớp: 732833

Contents

Bài thực hành số 4 – Tuần 16.....	122
Bài tập 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).	122
Bài tập 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.	125
Bài tập 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.	126
Bài tập 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.....	128
Bài tập 4.5. Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy.	131
Bài tập 4.6. Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy..	133
Bài tập 4.7. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set. .	134
Bài tập 4.8. Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.....	135
Bài tập 4.9. Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng danh sách kề sử dụng std::priority_queue.	136
Bài tập 4.10. Xây dựng một máy tìm kiếm (search engine) đơn giản.....	137
Bài tập 4.11. Hãy viết chương trình phân bố lính đứng ở các đoạn tường sao cho tổng số lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất.....	143

```
38 //Le Dinh Tri Tue - 0909 - 732833
39 struct components
PROBLEMS 45 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
23 2
10 46
13 26
5 56
20 97
1 22
17 63
13 42
7 75
15 87
6 48
7 16
19 40
11 47
19 14
15 67
10 6
23 86
10 36
20 23
7 12
13 14
22 13
2 75
242
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> []
```

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\" ; if (?) { g++ tempCodeRunnerFile.cpp
tempCodeRunnerFile } ; if (?) { .\tempCodeRunnerFile }

57 37
25 61
32 66
31 74
38 88
49 62
37 9
34 83
48 24
6 15
26 82
50 15
26 88
34 81
1 4
40 63
50 41
51 73
45 15
47 59
15 52
7 72
42 65
30 56
48 1
24 71
56 65
29 49
44 50
4 92
51 11
25 100
54 25
4 14
48 41
55 5
57 57

```

148

Bài tập 4.12. Cho một lược đồ gồm n cột chữ nhật liên tiếp nhau có chiều rộng bằng 1 và chiều cao lần lượt là các số nguyên không âm h1,h2,...,hn. Hãy xác định hình chữ nhật có diện tích lớn nhất có thể tạo thành từ các cột liên tiếp. 149

Bài tập 4.13. Cho một xâu nhị phân độ dài n. Hãy viết chương trình đếm số lượng xâu con chứa số ký tự 0 và số ký tự 1 bằng nhau. 152

Figure 1	122
Figure 2	122
Figure 3	123
Figure 4	123
Figure 5	124
Figure 6	125
Figure 7	125
Figure 8	126
Figure 9	127
Figure 10	128
Figure 11	128
Figure 12	129
Figure 13	129
Figure 14	130
Figure 15	131

Figure 16	132
Figure 17	133
Figure 18	134
Figure 19	134
Figure 20	135
Figure 21	135
Figure 22	136
Figure 23	136
Figure 24	137
Figure 25	141
Figure 26	142
Figure 27	143
Figure 28	147
Figure 29	148
Figure 30	149
Figure 31	152
Figure 32	154
Figure 33	154

Bài thực hành số 4 – Tuần 16

Copy code (kèm ảnh code và kết quả vào)

Bài tập 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)).

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, ... %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
/*
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))
*/
#include <iostream>
using namespace std;
//Le Dinh Tri Tue - 0909 - 732833
struct Node {
    int data;
    Node* next;
    Node(int data) {
        this->data = data;
    }
};

// push a new element to the beginning of the list
Node* prepend(Node* head, int data) {
    *****
    # YOUR CODE HERE #
    *****
    Node* tue = new Node(data);
    tue->next = head;
    return tue;
}
// print the list content on a line
void print(Node* head) {
```

Figure 99

```
Node(int data) {
    this->data = data;
    next = NULL;
}
};

// push a new element to the beginning of the list
Node* prepend(Node* head, int data) {
    *****
    # YOUR CODE HERE #
    *****
    Node* tue = new Node(data);
    tue->next = head;
    return tue;
}
// print the list content on a line
void print(Node* head) {
```

Figure 100

```

Node* temp = head;
while (true){
    std::cout << temp->data << " ";
    if (temp->next == NULL) break;
    temp = temp->next;
}
//Le Dinh Tri Tue - 0909 - 732833
// return the new head of the reversed list
Node* reverse(Node* head) {
    *****
    # YOUR CODE HERE #
    *****
    Node* prev = NULL;
    Node* current = head;
    Node* next = NULL;

    while (current != NULL) {

```

Figure 101

```

        while (current != NULL) {
            next = current->next;
            current->next = prev;
            prev = current;
            current = next;
        }

        head = prev;
        return head;
}
//Le Dinh Tri Tue - 0909 - 732833
int main() {
    int n, u;
    cin >> n;
    Node* head = NULL;
    for (int i = 0; i < n; ++i){
        cin >> u;
        head = prepend(head, u);
    }
}
```

Figure 102

```
}

cout << "Original list: ";
print(head);
cout << endl;
head = reverse(head);

cout << "Reversed list: ";
print(head);

return 0;
}
```

	Input	Expected	Got
✓	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	Original list: 50 12 7 6 4 2 7 5 4 -1 Reversed list: -1 4 5 7 2 4 6 7 12 50
✓	1 6	Original list: 6 Reversed list: 6	Original list: 6 Reversed list: 6
✓	15 2 3 -1 4 6 -7 12 5 7 12 4 76 2 5 54	Original list: 54 5 2 76 4 12 7 5 12 -7 6 4 -1 3 2 Reversed list: 2 3 -1 4 6 -7 12 5 7 12 4 76 2 5 54	Original list: 54 5 2 76 4 12 7 5 12 -7 6 4 -1 3 2 Reversed list: 2 3 -1 4 6 -7 12 5 7 12 4 76 2 5 54

Passed all tests! ✓

Figure 103

Bài tập 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.

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.

```
double area(Point a, Point b, Point c) {
    //*****
    # YOUR CODE HERE #
    *****/
}
```

trong đó, Point là kiểu được định nghĩa trước trong trình chấm như sau:

```
using Point = pair<double, double>;
```

For example:

Test	Result
cout << setprecision(2) << fixed; cout << area({1, 2}, {2.5, 10}, {15, -5.25}) << endl;	61.44

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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
/*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*/  
double area(Point a, Point b, Point c) {
    //*****
    # YOUR CODE HERE #
    *****/
    //Le Dinh Tri Tue - 0909 - 732833
    double res = 0.5*abs((b.first - a.first)*(c.second - a.second)
                        - (c.first - a.first)*(b.second - a.second));
}
```

Figure 104

```
*****  
//Le Dinh Tri Tue - 0909 - 732833  
double res = 0.5*abs((b.first - a.first)*(c.second - a.second)  
                    - (c.first - a.first)*(b.second - a.second));  
return res;  
}
```

	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	✓

Passed all tests! ✓

Figure 105

Bài tập 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.

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.

```
Vector cross_product(Vector a, Vector b) {
    //*****
    # YOUR CODE HERE #
    *****/
}
```

trong đó Vector là kiểu được định nghĩa sẵn trong trình chấm như sau:

```
using Vector = tuple<double, double, double>;
```

For example:

Test	Result
<pre>cout << setprecision(2) << fixed; Vector a {1.2, 4, -0.5}; Vector b {1.5, -2, 2.5}; Vector c = cross_product(a, b); cout << get<0>(c) << ' ' << get<1>(c) << ' ' << get<2>(c) << endl;</pre>	9.00 -3.75 -8.40

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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
/*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.*/
Vector cross_product(Vector a, Vector b) {
    //*****
    # YOUR CODE HERE #
    *****/
//Le Dinh Tri Tue - 0909 - 732833
double x = std::get<1>(a)*std::get<2>(b)
```

Figure 106

```
/*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.*/
Vector cross_product(Vector a, Vector b) {
    //*****
    # YOUR CODE HERE #
    *****/
//Le Dinh Tri Tue - 0909 - 732833
    double x = std::get<1>(a)*std::get<2>(b)
            - std::get<1>(b)*std::get<2>(a);
    double y = - std::get<0>(a)*std::get<2>(b)
            + std::get<0>(b)*std::get<2>(a);
    double z = - std::get<1>(a)*std::get<0>(b)
            + std::get<1>(b)*std::get<0>(a);
    return {x, y, z};
}
```

	Test	Expected	Got	
✓	cout << setprecision(2) << fixed; Vector a {1.2, 4, -0.5}; Vector b {1.5, -2, 2.5}; Vector c = cross_product(a, b); cout << get<0>(c) << ' ' << get<1>(c) << ' ' << get<2>(c) << endl;	9.00 -3.75 -8.40	9.00 -3.75 -8.40	✓
✓	cout << setprecision(2) << fixed; Vector a {-2.2, 4.5, -1.5}; Vector b {3.5, -7, 7.5}; Vector c = cross_product(a, b); cout << get<0>(c) << ' ' << get<1>(c) << ' ' << get<2>(c) << endl;	23.25 11.25 -0.35	23.25 11.25 -0.35	✓

Figure 107

Bài tập 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.

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, ... %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
/*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.*/
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
void print_vector(const vector<int> &a) {
    for (int v : a) cout << v << ' ';
    cout << endl;
}
//Le Dinh Tri Tue - 0909 - 732833
void delete_even(vector<int> &a) {
    // YOUR CODE HERE #
    *****
    for (std::vector<int>::iterator it = a.begin(); it != a.end(); ++it){
        if (*it % 2 == 0) {
```

Figure 108

```
        a.erase(it);
        it--;
    }
}
//Le Dinh Tri Tue - 0909 - 732833
void sort_decrease(vector<int> &a) {
    // YOUR CODE HERE #
    *****
    for (int i = 0; i < (int)a.size() - 1; ++i){
        for (int j = i + 1; j < (int)a.size(); ++j){
            if (a[i] < a[j]){
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}
```

Figure 109

```
//Le Dinh Tri Tue - 0909 - 732833
vector<int> merge_vectors(const vector<int> &a, const vector<int> &b) {
    //*****
    # YOUR CODE HERE #
    *****/
    std::vector<int> res = a;
    for (int i = 0; i < (int)b.size(); ++i){
        res.push_back(b[i]);
    }
    sort_decrease(res);
    return res;
}
//Le Dinh Tri Tue - 0909 - 732833
int main() {
    int m, n, u;
    std::vector<int> a, b;

    std::cin >> m >> n;
    //*****
}
```

Figure 110

```
for(int i = 0; i < m; i++){
    std::cin >> u;
    a.push_back(u);
}
for(int i = 0; i < n; i++){
    std::cin >> u;
    b.push_back(u);
}
//Le Dinh Tri Tue - 0909 - 732833
delete_even(a);
cout << "Odd elements of a: ";
print_vector(a);

delete_even(b);
cout << "Odd elements of b: ";
print_vector(b);

sort_decrease(a);
```

Figure 111

```

print_vector(b);

sort_decrease(a);
cout << "Decreasingly sorted a: ";
print_vector(a);

sort_decrease(b);
cout << "Decreasingly sorted b: ";
print_vector(b);
//Le Dinh Tri Tue - 0909 - 732833
vector<int> c = merge_vectors(a, b);
cout << "Decreasingly sorted c: ";
print_vector(c);

return 0;
}

```

	Input	Expected	Got
✓	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	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 Decreasingly sorted c: 35 13 9
✓	10 15 2 4 -7 2 5 7 13 9 43 55 12 3 65 32 2 4 675 76 21 57 87 321 54 76 -100	Odd elements of a: -7 5 7 13 9 43 55 Odd elements of b: 3 65 675 21 57 87 321 Decreasingly sorted a: 55 43 13 9 7 5 -7 Decreasingly sorted b: 675 321 87 65 57 21 3 Decreasingly sorted c: 675 321 87 65 57 55 43 21 13 9 7 5 3 -7	Odd elements of a: -7 5 7 13 9 Odd elements of b: 3 65 675 21 Decreasingly sorted a: 55 43 1 Decreasingly sorted b: 675 321 Decreasingly sorted c: 675 321

Passed all tests! ✓

Figure 112

Bài tập 4.5. Viết hàm void dfs(vector<list<int> > adj) thực hiện thuật toán DFS không sử dụng đệ quy.

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).

For example:

Test	Result
int n = 7;	1
vector<list<int>> adj;	2
adj.resize(n + 1);	4
adj[1].push_back(2);	7
adj[2].push_back(4);	3
adj[1].push_back(3);	5
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);	
dfs(adj);	

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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
/*
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).*/
void dfs(vector<list<int>>& adj) {
    stack<int> S;
    vector<bool> visited(adj.size());
    S.push(1);
```

Figure 113

Falling back to raw text area.

```
//Le Dinh Tri Tue - 0909 - 732833
if (!visited[currentVertex]) {
    if (currentVertex != 0) cout << currentVertex << " " << endl;
    visited[currentVertex] = true;
    auto it = adj[currentVertex].end();
    while (true) {
        bool flag = false;
        it--;
        if (it == adj[currentVertex].begin()) flag = true;
        int neighbor = *it;
        if (!visited[neighbor]) {
            S.push(neighbor);
        }
        if (flag == true) break;
    }
}
}
```

	Test	Expected	Got	
✓	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);	1 2 4 7 3 5	1 2 4 7 3 5	✓

Figure 114

Bài tập 4.6. Viết hàm void dfs(vector< list<int> > adj) thực hiện thuật toán BFS không sử dụng đệ quy..

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

For example:

Test	Result
int n = 7;	1
vector< list<int> > adj;	2
adj.resize(n + 1);	3
adj[1].push_back(2);	4
adj[2].push_back(4);	7
adj[1].push_back(3);	5
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);	

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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
//Le Dinh Tri Tue - 0909 - 732833
while (!Q.empty()) {
    int currentVertex = Q.front();
    Q.pop();
    //Le Dinh Tri Tue - 0909 - 732833
    if (!visited[currentVertex]) {
        cout << currentVertex << " " << std::endl;
        visited[currentVertex] = true;
        //Le Dinh Tri Tue - 0909 - 732833
        for (auto it = adj[currentVertex].begin(); it != adj[currentVertex].end(); ++it) {
```

Figure 115

```

        if (!visited[currentVertex]) {
            cout << currentVertex << " " << std::endl;
            visited[currentVertex] = true;
            //Le Dinh Tri Tue - 0909 - 732833
            for (auto it = adj[currentVertex].begin(); it != adj[currentVertex].end(); ++it) {
                int neighbor = *it;
                if (!visited[neighbor]) {
                    Q.push(neighbor);
                }
            }
        }
    }
}

```

	Test	Expected	Got	
✓	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);	1 2 3 4 7 5	1 2 3 4 7 5	✓
✓	int n = 10; vector< list<int> > adj; adj.resize(n + 1); adj[1].push_back(2); adj[2].push_back(1); adj[2].push_back(3); adj[3].push_back(2); adj[3].push_back(4); adj[4].push_back(3); adj[4].push_back(5); adj[5].push_back(4); adj[5].push_back(6); adj[6].push_back(5); adj[6].push_back(7); adj[7].push_back(6); adj[7].push_back(8); adj[8].push_back(7); adj[8].push_back(9); adj[9].push_back(8); adj[9].push_back(10); bfs(adj);	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	✓

Figure 116

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

```

/*viết các hàm thực hiện các phép giao và hợp của hai tập hợp được biểu diễn bằng set*/
template<class T>
set<T> set_union(const set<T> &a, const set<T> &b) {
    set<T> union_set;
    for(T tmp : a){
        union_set.insert(tmp);
    }
    for(T tmp : b){
        union_set.insert(tmp);
    }
    return union_set;
}
//Le Dinh Tri Tue - 0909 - 732833
template<class T>
set<T> set_intersection(const set<T> &a, const set<T> &b) {
    set<T> c;
    for(T v : a){
        int index = 0;

```

Figure 117

```
//Le Dinh Tri Tue - 0909 - 732833
template<class T>
set<T> set_intersection(const set<T> &a, const set<T> &b) {
    set<T> c;
    for(T v : a){
        int index = 0;
        for(T u : b){
            if(v == u){
                index++;
                break;
            }
        }
        if(index != 0) c.insert(v);
    }
    return c;
}
//Le Dinh Tri Tue - 0909 - 732833
```

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

Figure 118

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

```
/*Viết các hàm thực hiện các phép giao và hợp của hai tập hợp mờ được biểu diễn bằng map.  
Trong đó mỗi phần tử được gán cho một số thực trong đoạn [0..1] biểu thị độ thuộc  
của phần tử trong tập hợp, với độ thuộc bằng 1 nghĩa là phần tử chắc chắn thuộc vào  
tập hợp và ngược lại độ thuộc bằng 0 nghĩa là phần tử chắc chắn không thuộc trong tập hợp.  
Phép giao và hợp của 2 tập hợp được thực hiện trên các cặp phần tử bằng nhau của 2 tập hợp,  
với độ thuộc mới được tính bằng phép toán min và max của hai độ thuộc.*/
template<class T>
map<T, double> fuzzy_set_union(const map<T, double> &a, const map<T, double> &b) {
    map<T, double> c = a;
    for(const auto &e : b){
        if(c.count(e.first)){
            c[e.first] = max(e.second, c[e.first]);
        }
        else{
            c.insert(e);
        }
    }
    return c;
```

Figure 119

```

,
//Le Dinh Tri Tue - 0909 - 732833
template<class T>
map<T, double> fuzzy_set_intersection(const map<T, double> &a, const map<T, double> &b) {
    map<T, double> c;
    for(const auto &x : a){
        const auto it = b.find(x.first);
        if(it != b.end()){
            c[x.first] = min(x.second, it->second);
        }
    }
    return c;
}
//Le Dinh Tri Tue - 0909 - 732833

```

Test	Expected
<input checked="" type="checkbox"/> 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); map<int, double> c = fuzzy_set_union(a, b); map<int, double> d = fuzzy_set_intersection(a, b); cout << "Union: "; print_fuzzy_set(c); cout << "Intersection: "; print_fuzzy_set(d);	A = { (1, 0.2) (2, 0.5) (3, 1) (4, 0.6) (5, 1) B = { (1, 0.5) (2, 0.4) (4, 0.9) (5, 0.4) Union: { (1, 0.5) (2, 0.5) (3, 1) (4, 0.9) Intersection: { (1, 0.2) (2, 0.4) (4, 0.6)

Figure 120

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

```

/*Cài đặt thuật toán Dijkstra trên đồ thị vô hướng được biểu diễn bằng
danh sách kề sử dụng std::priority_queue
Cụ thể, bạn cần cài đặt hàm vector<int> dijkstra(const vector<vector<pair<int, int> >&adj)
nhận đầu vào là danh sách kề chứa các cặp pair<int, int> biểu diễn đỉnh kề
và trọng số tương ứng của cạnh. Đồ thị gồm n đỉnh được đánh số từ 0 tới n-1.
Hàm cần trả `vector<int>` chứa n phần tử lần lượt là khoảng cách đường đi ngắn nhất
từ đỉnh 0 tới các đỉnh 0, 1, 2, ..., n-1.*/
#include <iostream>
#include <queue>
#include <climits>
using namespace std;
//Le Dinh Tri Tue - 0909 - 732833
vector<int> dijkstra(const vector<vector<pair<int, int> >&adj) {
    priority_queue<pair<int,int>> s;
    vector<int> d(adj.size(), INT_MAX);
    d[0] = 0;
    s.push({0,0});
    while(!s.empty()) {

```

Figure 121

```

s.push({0, 0});
while(!S.empty()){
    int du = -S.top().first;
    int u = S.top().second;
    S.pop();
    if(du != d[u]) continue;
    for(auto e : adj[u]){
        int v = e.first;
        int c = e.second;
        if(d[v] > d[u] + c){
            d[v] = d[u] + c;
            S.push({-d[v], v});
        }
    }
}
//Le Dinh Tri Tue - 0909 - 732833
return d;
}

```

	Test	Expected	Got	
✓	<pre> int n = 9; vector< vector< pair<int, int> > > adj(n); auto add_edge = [&adj] (int u, int v, int w) { adj[u].push_back({v, w}); adj[v].push_back({u, w}); }; add_edge(0, 1, 4); add_edge(0, 7, 8); add_edge(1, 7, 11); </pre>	<pre> 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 </pre>	<pre> 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 </pre>	✓

Figure 122

Bài tập 4.10. Xây dựng một máy tìm kiếm (search engine) đơn giản.

CODE:

```
/*Xây dựng một máy tìm kiếm (search engine) đơn giản.
```

Cho N văn bản và Q truy vấn. Với mỗi truy vấn, cần trả về

văn bản khớp với truy vấn đó nhất. Sử dụng phương pháp tính điểm TF-IDF:*/

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
#define N 10000
```

```
vector<string> doc[N];
```

```
map<string, int> f[N];
```

```
map<string, int> df;
```

```
int maxf[N], n;
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
double TF(string t, int d){
```

```
if(f[d][t] == 0) return 0;  
return 0.5 + 0.5 * f[d][t] / maxf[d];  
}//Le Dinh Tri Tue - 0909 - 732833  
  
double IDF(string t){  
    return log2(1.0 * n / df[t]);  
}  
  
double score(string t, int d){  
    if(df.count(t) == 0) return 0;  
    return TF(t, d) * IDF(t);  
}//Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
    ios_base::sync_with_stdio(0);  
    cin.tie(0);  
    cout.tie(0);  
//Le Dinh Tri Tue - 0909 - 732833  
    cin >> n;  
    for(int i = 0; i < n; i++){  
        string t;  
        cin >> t;  
        string w;  
        map<string, int> d_f;  
//Le Dinh Tri Tue - 0909 - 732833  
        for(char c : t){  
            if (c != ',') w.push_back(c);  
            else {  
                d_f[w]++;
                if(d_f[w] == 1) df[w]++;
                doc[i].push_back(w);
                f[i][w]++;
                w = "";
            }
        }
    }
}
```

```

        continue;

    }

}

d_f[w]++;

if(d_f[w] == 1) df[w]++;

doc[i].push_back(w);

f[i][w]++;

for(auto it : f[i]){
    maxf[i] = max(maxf[i], it.second);
}

}

//Le Dinh Tri Tue - 0909 - 732833

int q;

cin >> q;

while(q--){
    vector<string> query;

    string t;

    cin >> t;

    string w;

    for(char c : t){
        if(c != ',') w.push_back(c);
        else{
            query.push_back(w);

            w = "";

            continue;
        }
    }

}

//Le Dinh Tri Tue - 0909 - 732833

query.push_back(w);

double tempscore = -1e9;

```

```
int document = 0;  
int d = 0;  
while (d < n){  
    double totalscore = 0;  
    for(string t : query){  
        totalscore += score(t, d);  
    }  
    d++;  
    if(totalscore > tempscore){  
        tempscore = totalscore;  
        document = d;  
    }  
}  
cout << document << "\n";  
}
```

OUTPUT:

The screenshot shows a terminal window with the following details:

- Code Snippets:** Lines 76 and 77 show code comments: //Le Dinh Tri Tue - 0909 - 732833 and QUERY.push_back(w).
- Terminal Tabs:** The tabs are labeled PROBLEMS (45), OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS.
- Command Line:** The command PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\" ; if (\$?) { & tempCodeRunnerFile } ; if (\$?) { .\tempCodeRunnerFile }
- Numbered Lines:** A vertical column of numbers on the left side of the terminal window, ranging from 71 down to 69.

Figure 123

The screenshot shows a terminal window with a dark background. At the top, there are tabs labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal area displays a list of numbers, each preceded by a line number from 75 to 53. The numbers are as follows:

Line Number	Value
75	75
76	//Le Dinh Tri Tue - 0909 - 732833
77	query.push_back(w);
78	
79	16
80	45
81	91
82	8
83	2
84	10
85	5
86	41
87	6
88	32
89	30
90	11
91	86
92	61
93	27
94	47
95	62
96	39
97	92
98	77
99	44
100	88
101	87
102	27
103	79
104	63
105	94
106	43
107	67
108	13
109	87
110	52
111	13
112	39
113	86
114	13
115	53

Figure 124

The screenshot shows a terminal window with the following content:

```
75
76     }  
77     //Le Dinh Tri Tue - 0909 - 732833  
78     query.push_back(w);  
PROBLEMS 45 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
13  
87  
52  
13  
39  
86  
13  
53  
9  
38  
41  
41  
35  
32  
57  
62  
70  
43  
52  
35  
94  
7  
32  
62  
92  
47  
16  
20  
40  
62  
93  
82  
42  
62  
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh>
```

Figure 125

Bài tập 4.11. Hãy viết chương trình phân bố lính đứng ở các đoạn tường sao cho tổng số lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất.

CODE:

```
/*

```

Bức tường bao quanh một lâu đài nọ được cấu thành từ n đoạn tường được đánh số từ 1 đến n

. Quân giặc lên kế hoạch tấn công lâu đài bằng cách gửi ai tên giặc đánh vào đoạn tường thứ i.

Để bảo vệ lâu đài có tất cả s lính. Do các đoạn tường có chất lượng khác nhau nên khả năng bảo vệ tại các đoạn tường cũng khác nhau. Cụ thể tại đoạn tường thứ i , mỗi lính có thể đẩy lùi tấn công của ki tên giặc.

Giả sử đoạn tường thứ i có xi lính. Khi đó nếu số tên giặc không vượt quá xi thì không có tên giặc nào lọt vào được qua đoạn tường này. Ngược lại sẽ có ai-xi tên giặc lọt vào lâu đài qua đoạn tường này. Yêu cầu hãy viết chương trình phân bổ lính đứng ở các đoạn tường sao cho tổng số lính là s và tổng số lượng tên giặc lọt vào lâu đài là nhỏ nhất.*/

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
void algo(){
```

```
    while(!now.empty() && s > 0){
```

```
        wall a = now.top();now.pop();
```

```
        if(a.ai <= a.ki){
```

```
            dich_da_giet += a.ai;
```

```
        } else {
```

```
            int now_enemy = a.ai - a.ki;
```

```
            now.push(wall(now_enemy,a.ki));
```

```
            dich_da_giet += a.ki;
```

```
}
```

```
    s -= 1;
```

```
}
```

```
}
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
struct wall{
```

```
    int ai;
```

```
int ki;

wall(int ai,int ki){

    this->ai = ai;
    this->ki = ki;
}

};

//Le Dinh Tri Tue - 0909 - 732833

struct compare{

    bool operator() (wall a, wall b){

        int ra, rb;

        if(a.ai <= a.ki) ra = a.ai;
        else ra = a.ki;

        if(b.ai <= b.ki) rb = b.ai;
        else rb = b.ki;

        return ra < rb;
    }
};

//Le Dinh Tri Tue - 0909 - 732833

int n,s;

priority_queue<wall,vector<wall>,compare> now;

int tong_dich=0,dich_da_giet=0;

void input(){

    cin >> n >> s;

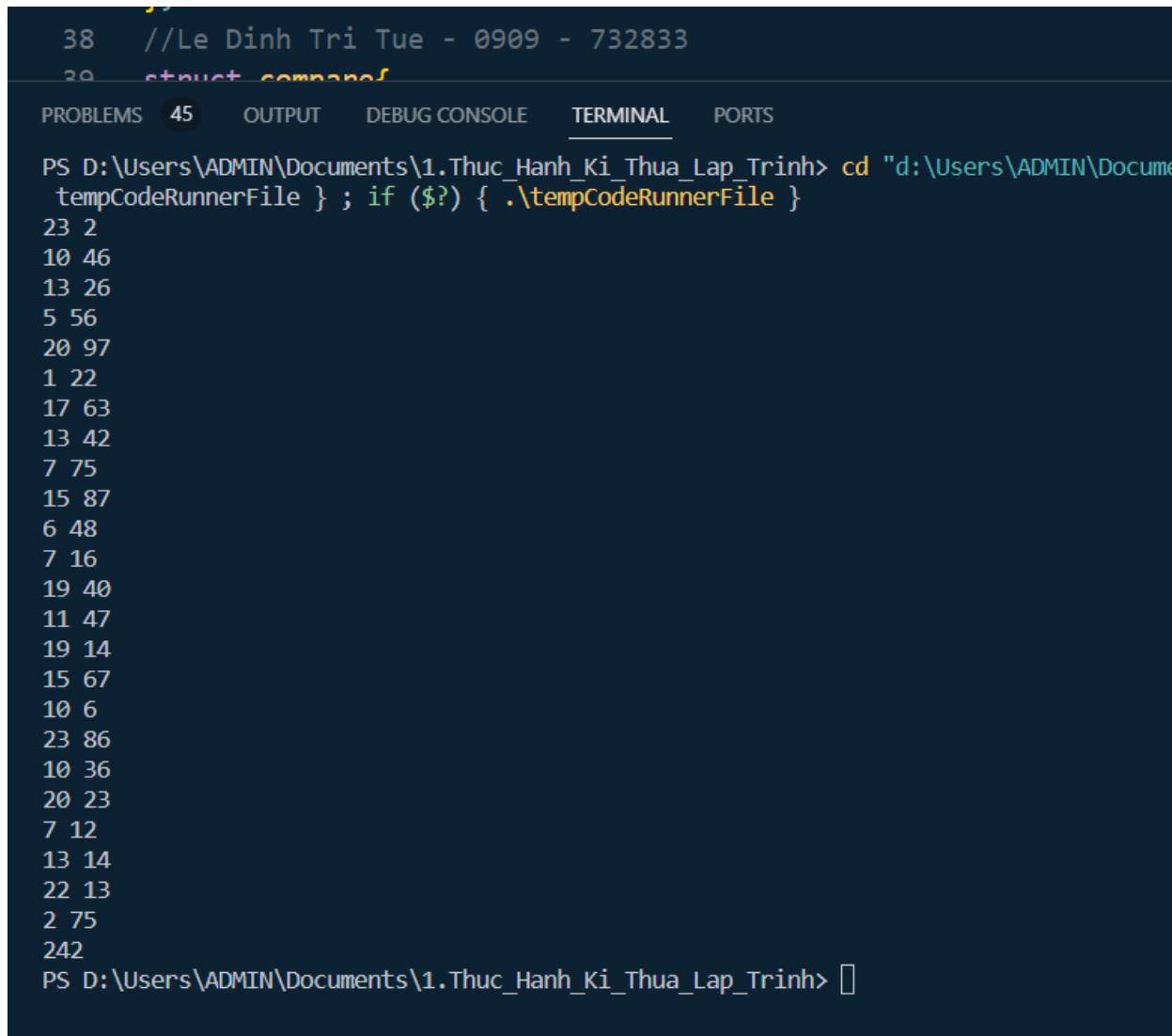
    int i = 0;

    while(i<n){

        int ai,ki;
```

```
cin >> ai >> ki;  
now.push(wall(ai,ki));  
tong_dich += ai;  
i++;  
}  
}  
  
//Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
    input();  
    algo();  
    cout << tong_dich - dich_da_giet;  
}
```

OUTPUT:



The screenshot shows a terminal window with the following content:

```
38 //Le Dinh Tri Tue - 0909 - 732833
39 struct components
PROBLEMS 45 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
23 2
10 46
13 26
5 56
20 97
1 22
17 63
13 42
7 75
15 87
6 48
7 16
19 40
11 47
19 14
15 67
10 6
23 86
10 36
20 23
7 12
13 14
22 13
2 75
242
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> []
```

Figure 126

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\" ; if ($?) { g++ tempCodeRunnerFile.cpp -tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }

57 37
25 61
32 66
31 74
38 88
49 62
37 9
34 83
48 24
6 15
26 82
50 15
26 88
34 81
1 4
40 63
50 41
51 73
45 15
47 59
15 52
7 72
42 65
30 56
48 1
24 71
56 65
29 49
44 50
4 92
51 11
25 100
54 25
4 14
48 41
55 5
57 57
```

Figure 127

```

29 49
44 50
4 92
51 11
25 100
54 25
4 14
48 41
55 5
57 57
42 44
52 98
3 87
42 79
7 13
35 34
38 87
36 38
36 52
7 67
3 59
44 42
40 5
26 16
51 68
49 67
22 76
17 85
21 38
14 35
24 52
506
PS D:\Users\ADMTN\Documents\1.Thuc_Hanh_Ki_Thua_Lan_Trinh> □

```

Figure 128

Bài tập 4.12. Cho một lược đồ gồm n cột chữ nhật liên tiếp nhau có chiều rộng bằng 1 và chiều cao lần lượt là các số nguyên không âm h1,h2,...,hn. Hãy xác định hình chữ nhật có diện tích lớn nhất có thể tạo thành từ các cột liên tiếp.

CODE:

```
/*
```

Cho một lược đồ gồm n cột chữ nhật liên tiếp nhau có chiều rộng bằng 1 và chiều cao lần lượt là các số nguyên không âm h1,h2,...,hn. Hãy xác định hình chữ nhật có diện tích lớn nhất có thể tạo thành từ các cột liên tiếp.

```
*/
```

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
int n;  
vector<int> graph;  
vector<vector<int>> neighbor;  
vector<int> value;  
  
void input(){  
  
    cin >> n;  
  
    graph.push_back(-1); //dat linh canh  
  
    for(int i = 0 ;i<n;i++){  
  
        int tmp;  
  
        cin >> tmp;  
  
        graph.push_back(tmp);  
  
    }  
  
    graph.push_back(-1); //linh canh  
  
}  
  
//Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
  
    input();  
  
    for(int i = 1;i<=n;i++){  
  
        int part = graph.at(i);  
  
        vector<int> a;  
  
        a.push_back(part);  
  
        //Le Dinh Tri Tue - 0909 - 732833  
  
        int now = i;  
  
        while(true){  
  
            if(graph.at(now - 1) < part){  
  
                break;  
            } else {  
  
                a.push_back(graph.at(now - 1));  
  
                now -= 1;  
            }  
        }  
        cout << a.size() << endl;  
    }  
}
```

```
}

now = i;

while(true){

    if(graph.at(now + 1) < part){

        break;

    } else {

        a.push_back(graph.at(now + 1));

        now += 1;

    }

}

value.push_back(part*a.size());

}

//Le Dinh Tri Tue - 0909 - 732833

sort(value.rbegin(),value.rend());

cout << value.at(0);

}
```

OUTPUT:

```

1  /*
2  Cho một lược đồ gồm n cột chữ nhật liên tiếp nhau có chiều rộng bằng 1 và chiều cao lần lượt
3  là các số nguyên không âm h1,h2,...,hn. Hãy xác định hình chữ nhật có diện tích lớn nhất có thể
4  tạo thành từ các cột liên tiếp.
5 */
6 #include<bits/stdc++.h>
7 using namespace std;
8 //Le Dinh Tri Tue - 0909 - 732833
9 int n;
10 vector<int> graph;
11 vector<vector<int>> neighbor;
12 vector<int> value;
13 void input(){
14     cin >> n;
15     graph.push_back(-1); //dat linh canh
16     for(int i = 0 ;i<n;i++){
17         int tmp;
18         cin >> tmp;
19         graph.push_back(tmp);
20     }
21     graph.push_back(-1); //linh canh
22 }
23 //Le Dinh Tri Tue - 0909 - 732833
24 int main(){
25     input();

```

PROBLEMS 45 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\" ; if ($?) {
tempCodeRunnerFile } ; if (?) { .\tempCodeRunnerFile }
case 1 = 45954
case 2 = 41684
case 3 = 500000000
case 4 = 48932400
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> []

```

Figure 129

Bài tập 4.13. Cho một xâu nhị phân độ dài n. Hãy viết chương trình đếm số lượng xâu con chứa số ký tự 0 và số ký tự 1 bằng nhau.

CODE:

```

/*
Cho một xâu nhị phân độ dài n. Hãy viết chương trình đếm số lượng xâu con chứa số ký tự 0 và số ký tự 1 bằng nhau.
*/
#include<bits/stdc++.h>

using namespace std;

const int MAX = 1000000;

string str;

int n, cnt;

int slg0[MAX]; // slg0[i] là đếm số lượng số 0 trong khoảng 0 tới i

int slg1[MAX]; // slg1[i] là đếm số lượng số 1 trong khoảng 0 tới i

```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
void input(){
```

```
    cnt = 0;
```

```
    cin >> str;
```

```
    n = str.length();
```

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

```
        if(i==0){
```

```
            if(str[i] == '0'){


```

```
                slg0[i] = 1;
```

```
                slg1[i] = 0;
```

```
            } else {


```

```
                slg0[i] = 0;
```

```
                slg1[i] = 1;
```

```
            }
```

```
            continue;
```

```
        }
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
        if(str[i] == '0'){


```

```
            slg0[i] = slg0[i-1] + 1;
```

```
            slg1[i] = slg1[i-1];
```

```
        } else {


```

```
            slg0[i] = slg0[i-1];
```

```
            slg1[i] = slg1[i-1] + 1;
```

```
        }
```

```
    }
```

```
}
```

```
//Le Dinh Tri Tue - 0909 - 732833
```

```
void solve(){


```

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


```

```
        for(int j=i+1; j<n; j = j + 2){


```

```
if(i==0){  
    if(slg0[j] == slg1[j]) cnt++;  
}  
else {  
    int c0 = slg0[j] - slg0[i-1];  
    int c1 = slg1[j] - slg1[i-1];  
    if(c0 == c1) cnt++;  
}  
}  
}  
}  
//Le Dinh Tri Tue - 0909 - 732833  
  
int main(){  
    ios_base::sync_with_stdio(false); cin.tie(NULL);  
    input();  
    solve();  
    cout << cnt;  
    return 0;  
}
```

OUTPUT:

```
tempCodeRunnerFile.cpp:11:1: error: expected unqualified id before numeric constant
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh"
tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
1001100110101101010001111111110000101000110101000100100100111101001100101010101101111011111011111101111
583
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh> []
```

Figure 130

Figure 131

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI THỰC HÀNH SỐ 5
MÔN KỸ THUẬT LẬP TRÌNH**

Sinh viên: Lê Đình Trí Tuệ

MSSV: 20210909

Mã lớp: 732833

Contents

Danh mục hình ảnh.....	157
Bài thực hành số 5 – Tuần 19.....	158
Bài tập 5.1. Tìm và sửa lỗi cú pháp của đoạn code sau:	158
Bài tập 5.2. Tìm và sửa các 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.	160
Bài tập 5.3. Dãy ngoặc đúng: Viết một chương trình nhận vào một dãy dấu ngoặc và kiểm tra xem dãy dấu ngoặc đóng mở đúng chưa.	162
Bài tập 5.4. Bài toán người du lịch	165
Bài tập 5.5. Năm nhuận: Một năm được coi là nhuận nếu nó chia hết cho 4 nhưng không chia hết cho 100, hoặc nó chia hết cho 400. Cho một danh sách các năm, kiểm tra xem có tồn tại năm nhuận trong danh sách đó hay không..	169
Bài tập 5.6. 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. 170	
Bài tập 5.7. 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).....	171
Bài tập 5.8. Cắt hình chữ nhật.....	173
Bài tập 5.9. Xây tháp.....	175

Danh mục hình ảnh

Figure 1: Bai 5.1.....	158
Figure 2: Bai 5.1.....	159
Figure 3: Bai 5.2.....	160
Figure 4: Bai 5.2.....	161
Figure 5: Bai 5.3.....	162
Figure 6: Bai 5.3.....	163
Figure 7: Bai 5.3.....	164
Figure 8: Bai 5.4.....	165
Figure 9: Bai 5.4.....	166
Figure 10: Bai 5.4	167
Figure 11: Bai 5.4	168
Figure 12: Bai 5.5	169
Figure 13: Bai 5.6	170
Figure 14: bai 5.7.....	171
Figure 15: Bai 5.7	172
Figure 16: Bai 5.8	173
Figure 17: bai 5.8.....	174
Figure 18: Bai 5.9	175
Figure 19: Bai 5.9	176
Figure 20: Bai 5.9	177

Bài thực hành số 5 – Tuần 19
Copy code (kèm ảnh code và kết quả vào)

Bài tập 5.1. Tìm và sửa lỗi cú pháp của đoạn code sau:

```
1  /*
2  Bài 5.1 : Tìm và sửa các lỗi cú pháp*/
3  #include <stdio.h>
4  //Le Dinh Tri Tue - 0909 - 732833
5
6  int x[100], mark[100], n;
7
8  void print(){
9      for (int i = 1; i <= n; ++i) printf("%d ", x[i]);
10     printf("\n");
11 }
12 //Le Dinh Tri Tue - 0909 - 732833
13
14 void process(int i) {
15     if (i > n){
16         print();
17         return;
18     }
19     for (int j = 1; j <= n; ++j)
```

Figure 132: Bai 5.1

```
17         return;
18     }
19     for (int j = 1; j <= n; ++j)
20         if (!mark[j]){
21             mark[j] = 1;
22             x[i] = j;
23             process(i+1);
24             mark[j] = 0;
25         }
26     }
27
28 int main() {
29     printf("Le Dinh Tri Tue - 0909 - 732833\n");
30     n = 5;
31     process(1);
32     return 0;
33 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Le Dinh Tri Tue - 0909 - 732833
1 2 3 4 5
1 2 3 5 4
1 2 4 3 5
1 2 4 5 3
1 2 5 3 4
1 2 5 4 3
1 3 2 4 5
1 3 2 5 4
1 3 4 2 5
1 3 4 5 2
1 3 5 2 4
1 3 5 4 2
1 4 2 3 5
```

Figure 133: Bai 5.1

Bài tập 5.2. Tìm và sửa các 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.

```

1  /*
2  Bài 5.2: Tìm và sửa các lỗi cú pháp
3  Bài toán cái túi: Cho một cái túi có sức chứa M và n đồ vật. Đồ vật thứ
4  i có khối lượng mi và giá trị vi. Cần chọn ra một số đồ vật để bỏ vào túi
5  sao cho tổng khối lượng không quá M và tổng giá trị là lớn nhất có thể.
6  Đoạn code sau đây giải bài toán cái túi bằng phương pháp duyệt nhánh cận.
7  Hãy tìm và sửa các lỗi cú pháp */
8 #include <iostream>
9 using namespace std;
0 //Le Dinh Tri Tue - 0909 - 732833
1 int n, M, m[100], v[100];
2 int x[100], best, sumV, sumM, all[100];
3
4 void init(){
5     for (int i = n; i >= 1; --i){
6         all[i] = all[i+1] + v[i];
7     }
8 }
9
0 void print() {
1     cout << best;
2 }
3 //Le Dinh Tri Tue - 0909 - 732833
4 void process(int i){
5     if (sumV + all[i] <= best || sumM > M) return ;
6     if (i > n){
7         best = sumV;
8         return ;

```

Figure 134: Bai 5.2

```
test.cpp > main()
24 void process(int i){
25     if (sumV + all[i] <= best || sumM > M) return ;
26     if (i > n){
27         best = sumV;
28         return ;
29     }
30     process(i+1);
31     sumM += m[i];
32     sumV += v[i];
33     process(i+1);
34     sumM -= m[i];
35     sumV -= v[i];
36 }
37
38 int main() {
39     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
40     cin >> n >> M;
41     for (int i = 1; i <= n; ++i)
42         cin >> m[i] >> v[i];
43     init();
44     process(1);
45     print();
46     return 0;
47 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
f (?) { g++ test.cpp -o test } ; if (?) { .\test }
Le Dinh Tri Tue - 0909 - 732833
3 10
3 4
4 5
5 6
11
PS D:\Users\ADMTR\Documents\1_Thuc_Hanh_Ki_Thuoc_Lan_Tren_Trinh\BTH05_732833_Tuan19_Tue\DT>
```

Figure 135: Bài 5.2

Bài tập 5.3. Dãy ngoặc đúng: Viết một chương trình nhận vào một dãy dấu ngoặc và kiểm tra xem dãy dấu ngoặc đóng mở đúng chưa.

```

1  /*
2  Bai 5.3: Dãy ngoặc đúng
3  Viết một chương trình nhận vào một dãy dấu ngoặc và kiểm tra
4  xem dãy dấu ngoặc đóng mở đúng chưa.
5  */
6 #include <iostream>
7 using namespace std;
8 #include <string.h>
9 #include <stack>
10 //Le Dinh Tri Tue - 0909 - 732833
11 int par(string str){
12     int a = str.length();
13     stack<char> S;
14     char x, y;
15     for (int i=0; i<a; i++){
16         x = str[i];
17         if (x == '(' || x == '[' || x == '{'){
18             S.push(x);
19         } // Le Dinh Tri Tue - 0909 - 732833
20         else {
21             if(S.size() == 0) return 0;
22             if (x == ')') {
23                 if (S.top() == '('){
24                     S.pop();
25                 }
26             }
27         }
28     }
29 }
```

Figure 136: Bai 5.3

```
26         else return 0;
27     }
28     else if (x == ']') {
29         if (S.top() == '['){
30             S.pop();
31         }
32         else return 0;
33     }
34     else if (x == '}') {
35         if (S.top() == '{'){
36             S.pop();
37         }
38         else return 0;
39     }
40 }
41 }
42 if (S.size() != 0){
43     return 0;
44 }
45 else return 1;
46 }
47
48 int main(){
49     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
50     int n;
```

Figure 137: Bai 5.3

```
47
48 int main(){
49     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
50     int n;
51     string str;
52
53     cin >> n;
54     for(int i=0; i<n; i++){
55         cin >> str;
56         cout << par(str) << endl;
57     }
58
59     return 0;
60 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\BTH05-732833-Tuan19-TueLDT>
if ($?) { .\test }
Le Dinh Tri Tue - 0909 - 732833
2
()[]()()
1
()()
0
```

Figure 138: Bai 5.3

Bài tập 5.4. Bài toán người du lịch

```
/*
Bai 5.4: Bài toán người du lịch*/
#include <bits/stdc++.h>
using namespace std;
int m, n, Smin = 1000000;
long long S = 0;
int cmin = 100000000;
int x[100];
int c[100][100];
vector<int> flag(100, false);
//Le Dinh Tri Tue - 0909 - 732833
void TRY(int k)
{
    for (int i = 2; i <= n; i++)
    {
        if (flag[i] == false && c[x[k - 1]][i] != -1)
        {
            flag[i] = true;
            x[k] = i;
            S = S + c[x[k - 1]][i];
            if (k == n)
                cout << S << endl;
            TRY(k + 1);
            flag[i] = false;
            x[k] = 0;
            S -= c[x[k - 1]][i];
        }
    }
}
```

Figure 139: Bai 5.4

```
19         x[k] = i;
20         S = S + c[x[k - 1]][i];
21         if (k == n)
22         {
23             if (S + c[i][1] < Smin && c[i][1] != -1)
24                 Smin = S + c[i][1];
25             }
26             else if (S + cmin * (n - k + 1) < Smin)
27             {
28                 TRY(k + 1);
29             }
30             flag[i] = false;
31             S = S - c[x[k - 1]][i];
32         }
33     }
34 }
35
36 int main()
37 {
38     printf("Le Dinh Tri Tue\n");
39     printf("20210909\n\n");
```

Figure 140: Bai 5.4

```
35     }
36     int main()
37     {
38         printf("Le Dinh Tri Tue\n");
39         printf("20210909\n\n");
40         int a, b;
41         cin >> n >> m;
42         for (int i = 1; i <= n; i++)
43             for (int j = 1; j <= n; j++)
44             {
45                 if (i == j)
46                     c[i][j] = 0;
47                 else
48                     c[i][j] = -1;
49             }
50         for (int i = 0; i < m; i++)
51         {
52             cin >> a >> b;
53             cin >> c[a][b];
54             if (c[a][b] < cmin)
55                 cmin = c[a][b];
```

Figure 141: Bai 5.4

```
49 }  
50     for (int i = 0; i < m; i++)  
51 {  
52     cin >> a >> b;  
53     cin >> c[a][b];  
54     if (c[a][b] < cmin)  
55         cmin = c[a][b];  
56 }  
57 x[1] = 1;  
58 flag[1] = true;  
59 TRY(2);  
60 cout << Smin;  
61 return 0;  
62 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
if ($?) { .\test }  
Le Dinh Tri Tue  
20210909  
  
4 4  
1 3 2  
2 4 2  
1 4 3  
2 3 1  
1000000
```

Figure 142: Bai 5.4

Bài tập 5.5. Năm nhuận: Một năm được coi là nhuận nếu nó chia hết cho 4 nhưng không chia hết cho 100, hoặc nó chia hết cho 400. Cho một danh sách các năm, kiểm tra xem có tồn tại năm nhuận trong danh sách đó hay không..

```

3  Bài 5.5: Năm nhuận
4  Một năm được coi là nhuận nếu nó chia hết cho 4 nhưng không
5  chia hết cho 100, hoặc nó chia hết cho 400. Cho một danh sách các năm,
6  kiểm tra xem có tồn tại năm nhuận trong danh sách đó hay không.*/
7  #include <bits/stdc++.h>
8  using namespace std;
9  //Le Dinh Tri Tue - 0909 - 732833
10 int main(){
11     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
12     int n;
13     cin >> n;
14     bool found = false;
15     while(n--){
16         int a;
17         cin >> a;
18         if (a % 4 == 0 && a % 100 != 0)
19             found = true;
20         else if(a % 400 == 0){
21             found = true;
22         }
23     }
24     if (found) cout << "Yes";
25     else cout << "No";
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

19-TueLDT\" ; if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
Le Dinh Tri Tue - 0909 - 732833
4
2010 2011 2012 2013
Yes

```

Figure 143: Bai 5.5

Bài tập 5.6. 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.

```

3  Bai 5.6 Tổng kết
4  Một lớp có n sinh viên. Sinh viên thứ i có điểm tổng kết là
5  ai theo thang điểm 10. Để đánh giá chất lượng dạy học,
6  giảng viên muốn biết có bao nhiêu bạn đạt điểm A, B, C, D, F.*/
7 #include <bits/stdc++.h>
8 using namespace std;
9 //Le Dinh Tri Tue - 0909 - 732833
10 int main(){
11     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
12     float n;
13     cin >> n;
14     int A = 0, B = 0, C = 0, D = 0, F = 0;
15     while(n--){
16         float a;
17         cin >> a;
18         if (7 <= a && a < 8.5) ++B;
19         else if (5.5 <= a && a < 7) ++C;
20         else if (8.5 <= a) ++A;
21         else if (4 <= a && a < 5.5) ++D;
22         else if (a < 4) ++F;
23     }
24     cout << A << " " << B << " " << C << " " << D << " " << F;
25 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

19-TueLDT\" ; if ($?) { g++ test.cpp -o test } ; if ($?) { .\test }
Le Dinh Tri Tue - 0909 - 732833
10
1 2 3 4 5 6 7 8 9 10
2 2 1 2 3
PS D:\Users\ADMTN\Documents\1_Thuc_Hanh_Ki_Thuoc_Lan_Trinh\BTH05-732833-Tuan19-TueLDT>
```

Figure 144: Bai 5.6

Bài tập 5.7. 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 k 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 k nhà này và xin ai,j tiền của nhà thứ j. Nếu thôn i nghèo, họ sẽ đi từng nhà trong số k 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).

```
/*
Bai 5.7: 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 k 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 k nhà này và xin ai,j tiền của nhà thứ j. Nếu thôn i nghèo, họ sẽ đi từng nhà trong số k 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)*/

#include <bits/stdc++.h>
using namespace std;
//Le Dinh Tri Tue - 0909 - 732833
int main(){
    cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
    int n;
    cin >> n;
    int ans = 0, sum = 0;
    while(n--){
        int k, t;
        cin >> k >> t;
        if(t == 1){
            while(k--){
                int a;
                cin >> a;
                sum += a;
        }
    }
}
```

Figure 145: bai 5.7

```
test.cpp
18     int k, t;
19     cin >> k >> t;
20     if(t == 1){
21         while(k--){
22             int a;
23             cin >> a;
24             sum += a;
25             ans = max(ans, -sum);
26         }
27     }
28     else if(t == -1){
29         while(k--){
30             int a;
31             cin >> a;
32             sum -= a;
33             ans = max(ans, -sum);
34         }
35     }
36 }
37 cout << ans;
38 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Le Dinh Tri Tue - 0909 - 732833
2
3 1
1 2 3
3 -1
1 4 7
6
```

Figure 146: Bai 5.7

Bài tập 5.8. Cắt hình chữ nhật.

```
2  /*
3   Bai 5.8 Cắt hình chữ nhật*/
4  #include <bits/stdc++.h>
5  using namespace std;
6  // Le Dinh Tri Tue - 0909 - 732833
7  int w, h;
8  int table[601][601];
9  void init() {
10     for (int i=1; i<=h; i++) {
11         for (int j=1; j<=w; j++) {
12             table[i][j] = i*j;
13         }
14     }
15 }
16 int main()
17 {
18     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
19     ios::sync_with_stdio(false);
20     cin.tie();
21     int m;
22     cin >> w >> h;
23     cin >> m;
24     init();
25     for (int i=0; i<m; i++) {
```

Figure 147: Bai 5.8

The screenshot shows a code editor window with a dark theme. The file is named test.cpp. The code implements a dynamic programming algorithm to solve a problem involving horizontal and vertical cuts. It includes input reading, a DP table initialization phase, and a main loop for calculating minimum waste values. The output terminal shows sample inputs and their corresponding outputs.

```
test.cpp > ⚡ main()
24     init();
25     for (int i=0; i<m; i++) {
26         int tmp1, tmp2;
27         cin >> tmp1 >> tmp2;
28         table[tmp2][tmp1] = 0;
29     }
30     // Le Dinh Tri Tue - 0909 - 732833
31     //dp
32     for (int i=1; i<=h; i++) {
33         for (int j=1; j<=w; j++) {
34             int minWaste = table[i][j];
35             // horizontal cut
36             for(int k=1; k<=i; k++) {
37                 minWaste = min(minWaste, table[k][j] + table[i-k][j]);
38             }
39             // vertical cut
40             for (int k=1; k<=j; k++) {
41                 minWaste = min(minWaste, table[i][k] + table[i][j-k]);
42             }
43             table[i][j] = minWaste;
44         }
45     }
46     cout << table[h][w] << endl;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Le Dinh Tri Tue - 0909 - 732833
21 11
4
10 4
6 2
7 5
15 10
10
```

Figure 148: bai 5.8

Bài tập 5.9. Xây tháp.

```
2  /*
3   Bài tập 5.9: Xây tháp*/
4  #include <bits/stdc++.h>
5  using namespace std;
6  //Le Dinh Tri Tue - 0909 - 732833
7  typedef struct {
8      int x, y, z;
9  } block;
10 int n;
11 block a[100];
12 int maxh[100];
13
14 void input(){
15     cin >> n;
16     if (n == 0) exit(0);
17     int x, y, z;
18     for (int i = 1; i <= n; i++){
19         cin >> x >> y >> z;
20         a[3 * i - 2].x = x;
21         a[3 * i - 2].y = y;
22         a[3 * i - 2].z = z;
23         a[3 * i - 1].x = y;
24         a[3 * i - 1].y = z;
25         a[3 * i - 1].z = x;
26         a[3 * i].x = z;
27         a[3 * i].y = x;
28         a[3 * i].z = y;
29     }
    . . .
}
```

Figure 149: Bai 5.9

```
33 }
34 //Le Dinh Tri Tue - 0909 - 732833
35 int dp(int i){
36     if (maxh[i] != 0) return maxh[i];
37     maxh[i] = a[i].z;
38     for(int j = 1; j <= 3*n; j++){
39         if (a[i].x < a[j].x && a[i].y < a[j].y ||
40             a[i].x < a[j].y && a[i].y < a[j].x){
41             maxh[i] = max (maxh[i], a[i].z + dp(j));
42         }
43     }
44     return maxh[i];
45 }
46 //Le Dinh Tri Tue - 0909 - 732833
47 int main(){
48     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
49     int cnt = 1;
50     while(1){
51         int res = 0;
52         input();
53         for(int i = 1; i <= 3 * n; i++){
54             res = max(res, dp(i));
55         }
56         printf("Case %d: maximum height = %d\n", cnt++, res);
57     }
58     return 0;
59 }
```

Figure 150: Bai 5.9

```
45     }
44     return maxh[i];
45 }
46 //Le Dinh Tri Tue - 0909 - 732833
47 int main(){
48     cout << "Le Dinh Tri Tue - 0909 - 732833" << endl;
49     int cnt = 1;
50     while(1){
51         int res = 0;
52         input();
53         for(int i = 1; i <= 3 * n; i++){
54             res = max(res, dp(i));
55         }
56         printf("Case %d: maximum height = %d\n", cnt++, res);
57     }
58     return 0;
59 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Users\ADMIN\Documents\1.> cd "d:\Users\ADMIN\Documents\1.Thuc_Hanh_Ki_Thua_Lap_Trinh\E$?") { .\test }
Le Dinh Tri Tue - 0909 - 732833
2
1 2 3
4 5 6
Case 1: maximum height = 14
3
5 7 2
2 5 8
7 5 8
Case 2: maximum height = 21
```

Figure 151: Bai 5.9