

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



BÁO CÁO THỰC HÀNH
KỸ THUẬT LẬP TRÌNH – IT3040
Bài số 1 – Tuần 9: Con trỏ và cấp phát động

Sinh viên thực hiện: Nguyễn Hoàng Lâm

Mã số sinh viên: 20210517

MỤC LỤC

DANH MỤC HÌNH ẢNH	3
BÀI THỰC HÀNH SỐ 1 – TUẦN 9	4
Phần 1. Thực hành về con trỏ.....	4
Bài 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 1.2. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: <code>int a[7]= {13, -355, 235, 47, 67, 943, 1222}</code>	5
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.	5
Phần 2. Con trỏ và mảng	6
Bài 1.4. Viết hàm <code>countEven(int*, int)</code> nhận một mảng số nguyên và kích thước của mảng, trả về số lượng số chẵn trong mảng.	6
Bài 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.....	7
Bài 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.	7
Phần 3. Cấp phát động.....	9
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.....	9
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.	10
Phần 4. Bài tập về nhà.....	11
Bài 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước. Ví dụ: 11	
Bài 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.	17

DANH MỤC HÌNH ẢNH

Hình 1. Bài 1.1_LamNH_0517 - 1.....	4
Hình 2. Bài 1.1_LamNH_0517 - 2.....	4
Hình 3. Bài 1.2_LamNH_0517.....	5
Hình 4. Bài 1.3_LamNH_0517 - 1.....	5
Hình 5. Bài 1.3_LamNH_0517 - 2.....	6
Hình 6. Bài 1.4_LamNH_0517.....	6
Hình 7. Bài 1.5_LamNH_0517.....	7
Hình 8. Bài 1.6_LamNH_0517 - 1.....	8
Hình 9. Bài 1.6_LamNH_0517 - 2.....	8
Hình 10. Bài 1.7_LamNH_0517 - 1.....	9
Hình 11. Bài 1.7_LamNH_0517 - 2.....	9
Hình 12. Bài 1.8_LamNH_0517 - 1.....	10
Hình 13. Bài 1.8_LamNH_0517 - 2.....	10
Hình 14. Bài 1.8_LamNH_0517 - 3.....	11
Hình 15. Bài 1.8_LamNH_0517 - 4.....	11
Hình 16. Bài 1.9_LamNH_0517.....	12
Hình 17. Bài 1.9_LamNH_0517 - TestCase 1	13
Hình 18. Bài 1.9_LamNH_0517 - TestCase 2	13
Hình 19. Bài 1.9_LamNH_0517 - TestCase 3 - 1	13
Hình 20. Bài 1.9_LamNH_0517 - TestCase 3 - 2	14
Hình 21. Bài 1.9_LamNH_0517 - TestCase 4 - 1	14
Hình 22. Bài 1.9_LamNH_0517 - TestCase 4 - 2	15
Hình 23. Bài 1.9_LamNH_0517 - TestCase 4 - 3	15
Hình 24. Bài 1.9_LamNH_0517 - TestCase 4 - 4	16
Hình 25. Bài 1.9_LamNH_0517 - TestCase 4-5	16
Hình 26. Bài 1.10_LamNH_0517 - Code 1	17
Hình 27. Bài 1.10_LamNH_0517 - Code 2	18
Hình 28. Bài 1.10_LamNH_0517 – Code - 3	18
Hình 29. Bài 1.10_LamNH_0517 - TestCase 1	19
Hình 30. Bài 1.10_LamNH_0517 - TestCase 2	19
Hình 31. Bài 1.10_LamNH_0517 - TestCase 3	20
Hình 32. Bài 1.10_LamNH_0517 - TestCase 4	20

BÀI THỰC HÀNH SỐ 1 – TUẦN 9

Phần 1. Thực hành về con trỏ

Bài 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.

Bài làm

Chương trình:

```

Answer: (penalty regime: 10, 20, ... %)
1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2     Bài 1.1. Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt
3     trỏ tới từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.*/
4  #include <stdio.h>
5  int main(){
6     int x, y, z;
7     int* ptr;
8     printf("Enter three integers: ");
9     scanf("%d %d %d", &x, &y, &z); // Nhập 3 số nguyên x, y, z
10    printf("\nThe three integers are: \n");
11    ptr = &x; // Gán địa chỉ của biến x cho con trỏ ptr
12    printf("x = %d\n", *ptr);
13    /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
14    ptr = &y; // Gán địa chỉ của biến y cho con trỏ ptr
15    printf("y = %d\n", *ptr);
16    ptr = &z; // Gán địa chỉ của biến z cho con trỏ ptr
17    printf("z = %d\n", *ptr);
18    /******
19
20    return 0;
21 }
22 // Nguyễn Hoàng Lâm - 0517 - 732833
    
```

Hình 1. Bài 1.1_LamNH_0517 - 1

```

19
20     return 0;
21 }
22 // Nguyễn Hoàng Lâm - 0517 - 732833
        
```

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 2. Bài 1.1_LamNH_0517 - 2


```

19 printf("Once again, here are the values of x, y, and z:\n");
20 printf("%d %d %d\n", x, y, z); // // In ra giá trị mới của x, y, và z
21 return 0;
22 }
23 // Nguyễn Hoàng Lâm - 0517 - 732833
24
25

```

Precheck

	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	✓

Passed all tests! ✓

Hình 5. Bài 1.3_LamNH_0517 - 2

Phần 2. Con trỏ và 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.

Bài làm

Chương trình:

```

1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2     Bài 1.4. Viết hàm countEven(int*, int) nhận một mảng số nguyên và kích thước
3     của mảng, trả về số lượng số chẵn trong mảng.*/
4  int counteven(int* arr, int size){
5     int count = 0; // Khởi tạo biến count để đếm số lượng phần tử chẵn
6
7     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
8     for(int i = 0; i<size; i++){
9         if(*(arr+i)%2 == 0) count ++; // Tăng biến count lên nếu phần tử là số chẵn
10    }
11    /******/
12
13    return count; // Trả về số lượng phần tử chẵn trong mảng
14 }
15 // Nguyễn Hoàng Lâm - 0517 - 732833
16

```

Precheck

	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 6. Bài 1.4_LamNH_0517

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

Bài làm

Chương trình:

```
6  /* Nguyễn Hoàng Lâm - 0517 - 732833
7     Bài 1.5. Viết hàm trả về con trỏ tới giá trị lớn nhất
8     của một mảng các số double. Nếu mảng rỗng hãy trả về NULL. */
9  double* maximum(double* a, int size){
10     double *max; // Khởi tạo con trỏ 'max' lưu trữ phần tử lớn nhất
11     max = a; // Gán con trỏ 'max' ban đầu bằng con trỏ 'a'
12     if (a == NULL) return NULL; // Kiểm tra con trỏ 'a' trỏ đến mảng hợp lệ không
13
14     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
15     for(int i = 0; i < size; i++){
16         if(*(a+i) > *max) max = a+i; // Cập nhật con trỏ 'max' nếu tìm thấy phần tử lớn hơn
17     }
18     /******
19
20     return max; // Trả về con trỏ đến phần tử lớn nhất trong mảng
21 }
22 // Nguyễn Hoàng Lâm - 0517 - 732833
```

Precheck

	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 7. Bài 1.5_LamNH_0517

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

Bài làm

Chương trình:

Nguyễn Hoàng Lâm - 20210517

```

1 ▽ /* Nguyễn Hoàng Lâm - 0517 - 732833
2   Bài 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. */
3 ▽ void reversearray(int arr[], int size){
4   int l = 0, r = size - 1, tmp;
5   /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
6   while(r > l){
7       // Hai chỉ số 'l' và 'r' duyệt mảng từ hai đầu và đổi chỗ các phần tử.
8       tmp = arr[l];
9       arr[l] = arr[r];
10      arr[r] = tmp;
11      l++;
12      r--;
13  }
14 ▽ /******
15  }
16 ▽ void ptr_reversearray(int *arr, int size){
17   int l = 0, r = size - 1, tmp;
18   /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
19 ▽ while(r > l){
20   // Hai con trỏ 'arr + l' và 'arr + r' duyệt mảng từ hai đầu và đổi chỗ các phần tử.
21   tmp = *(arr + l);
22   *(arr + l) = *(arr + r);

```

Hình 8. Bài 1.6_LamNH_0517 - 1

```

16 ▽ void ptr_reversearray(int *arr, int size){
17   int l = 0, r = size - 1, tmp;
18   /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
19 ▽ while(r > l){
20   // Hai con trỏ 'arr + l' và 'arr + r' duyệt mảng từ hai đầu và đổi chỗ các phần tử.
21   tmp = *(arr + l);
22   *(arr + l) = *(arr + r);
23   *(arr + r) = tmp;
24   l++;
25   r--;
26   }
27 ▽ /******
28   }
29 // Nguyễn Hoàng Lâm - 0517 - 732833

```

Precheck

	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 9. Bài 1.6_LamNH_0517 - 2

Phần 3. Cấp phát động

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.

Bài làm

Phân tích: Chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.

Chương trình:

```

1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2  Bài 1.7. Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím.
3  Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp. */
4  #include <stdio.h>
5  int *a;
6  int n, tmp;
7  int main(){
8      printf("Enter the number of elements: ");
9      scanf("%d", &n);
10
11      // ALLOCATE MEMORY /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
12      a = new int[n]; // Cấp phát bộ nhớ động cho mảng 'a' kích thước n
13      if(a == NULL) return 0; // Thoát chương trình nếu cấp phát không thành công (Không đủ bộ nhớ)
14      /******
15      for(int i = 0; i < n; i++)
16          scanf("%d", a + i); // Nhập giá trị cho từng phần tử của mảng
17
18      printf("The input array is: \n");
19      for(int i = 0; i < n; i++)
20          printf("%d ", *(a + i)); // In ra mảng ban đầu
21      printf("\n");
22  */

```

Hình 10. Bài 1.7_LamNH_0517 - 1

```

23  // SORT ARRAY /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
24  for(int i = 0; i < n; i++){ // Sắp xếp mảng 'a' bằng thuật toán sắp xếp nổi bọt.
25      for(int j = i; j < n; j++){
26          if(*(a+i) > *(a+j)){
27              tmp = *(a+i);
28              *(a+i) = *(a+j);
29              *(a+j) = tmp;
30          }
31      }
32  }
33  /******
34
35  printf("The sorted array is: \n");
36  for(int i = 0; i < n; i++)
37      printf("%d ", *(a + i)); // In ra mảng sau khi sắp xếp
38  printf("\n");
39
40  delete [] a; // Giải phóng bộ nhớ đã cấp phát cho mảng 'a'
41  return 0;
42  }
43  // Nguyễn Hoàng Lâm - 0517 - 732833
44  */

```

Precheck

	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 11. Bài 1.7_LamNH_0517 - 2

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.

Bài làm

Phân tích: 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`.

Chương trình:

```
1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2     Bài 1.8. Viết chương trình nhập vào một ma trận 2 chiều kích thước m*n với m và n nhập từ bàn phím.
3     Đưa ra tổng các phần tử chẵn của ma trận đó. */
4  #include <stdio.h>
5  void allocate_mem(int ***mt, int m, int n){
6     // Allocate memory for the matrix
7     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
8     *mt = new int*[m]; // Cấp phát mảng con trỏ chứa m hàng
9     for(int i = 0; i<m; i++){
10        (*mt)[i] = new int[n]; // Cấp phát mảng số nguyên có n cột cho từng hàng của ma trận
11    }
12    /******/
13 }
14 void input(int **mt, int m, int n){
15     // Input elements of the matrix
16     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
17     for(int i = 0; i<m; i++){
18         for(int j = 0; j<n; j++){
19             printf("mt[%d][%d] = ", i, j);
20             scanf("%d", &mt[i][j]); // Nhập giá trị cho từng phần tử của ma trận
21         }
22     }
```

Hình 12. Bài 1.8_LamNH_0517 - 1

```
21     }
22 }
23 /******/
24 }
25 void output(int **mt, int m, int n){
26     // Print all elements of the matrix
27     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
28     for(int i = 0; i<m; i++){
29         for(int j = 0; j<n; j++){
30             printf("%d ", mt[i][j]); // In giá trị của từng phần tử trong ma trận
31         }
32         printf("\n"); // In xuống dòng sau mỗi hàng của ma trận
33     }
34     /******/
35 }
36 int process(int **mt, int m, int n){
37     int tong = 0; // Khởi tạo biến tong để tính tổng các phần tử chẵn trong ma trận
38     // Calculate the sum of all even elements in the matrix
39     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
40     for(int i = 0; i<m; i++){
41         for(int j = 0; j<n; j++){
42             if(mt[i][j]%2 == 0) tong+= mt[i][j]; // Cộng giá trị của phần tử chẵn vào biến tong
```

Hình 13. Bài 1.8_LamNH_0517 - 2

```

40 ▾   for(int i = 0; i<m; i++){
41 ▾       for(int j = 0; j<n; j++){
42 ▾           if(mt[i][j]%2 == 0) tong+= mt[i][j]; // Cộng giá trị của phần tử chẵn vào biến tong
43 ▾       }
44 ▾   }
45 ▾   /*****/
46 ▾   return tong; // Trả về tổng các phần tử chẵn trong ma trận
47 ▾ }
48 ▾ void free_mem(int **mt, int m, int n){
49 ▾     // Free memory
50 ▾     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
51 ▾     for(int i = 0; i<m; i++){
52 ▾         delete[] mt[i]; // Giải phóng bộ nhớ cho từng hàng của ma trận
53 ▾     }
54 ▾     delete[] mt; // Giải phóng bộ nhớ cho mảng con trỏ
55 ▾     /*****/
56 ▾ }
57 ▾ int main(){
58 ▾     int m, n, **mt;
59 ▾     printf("Enter m, n = ");
60 ▾     scanf("%d%d", &m, &n);
61 ▾     allocate_mem(&mt, m, n); // Cấp phát bộ nhớ cho ma trận

```

Hình 14. Bài 1.8_LamNH_0517 - 3

```

49 ▾     // Free memory
50 ▾     /*YOUR CODE HERE - Nguyễn Hoàng Lâm - 0517 - 732833*/
51 ▾     for(int i = 0; i<m; i++){
52 ▾         delete[] mt[i]; // Giải phóng bộ nhớ cho từng hàng của ma trận
53 ▾     }
54 ▾     delete[] mt; // Giải phóng bộ nhớ cho mảng con trỏ
55 ▾     /*****/
56 ▾ }
57 ▾ int main(){
58 ▾     int m, n, **mt;
59 ▾     printf("Enter m, n = ");
60 ▾     scanf("%d%d", &m, &n);
61 ▾     allocate_mem(&mt, m, n); // Cấp phát bộ nhớ cho ma trận
62 ▾     input(mt, m, n); // Nhập giá trị cho ma trận
63 ▾     output(mt, m, n); // In ra giá trị của ma trận
64 ▾     printf("The sum of all even elements is %d", process(mt, m, n));
65 ▾     free_mem(mt, m, n); // Giải phóng bộ nhớ của ma trận
66 ▾     return 0;
67 ▾ }
68 ▾ // Nguyễn Hoàng Lâm - 0517 - 732833
69 ▾
70 ▾

```

Precheck

Check

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

Passed all tests! ✓

Hình 15. Bài 1.8_LamNH_0517 - 4

Phần 4. Bài tập về nhà

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

Input: 1 3 4 2

Output:

1

1 3

1 3 4

1 3 4 2

3

3 4

3 4 2

4

4 2

2

Bài làm

Chương trình (Trên Visual Code Studio):

```
1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2  |   Bài 1.9. (BTVN) Viết chương trình in ra tất cả các dãy con của một dãy cho trước. */
3  #include<bits/stdc++.h>
4  using namespace std;
5  int main(){
6      int n; // Lưu trữ kích thước của mảng
7      cout << "Bai1.09_LamNH_0517" << endl << "input = ";
8      cin >> n; // Nhập kích thước mảng
9
10     int* A = new int[n]; // Cấp phát bộ nhớ động mảng A có kích thước n
11     ...
12     for(int i = 0; i < n; i++){
13         cin >> A[i]; // Nhập giá trị cho từng phần tử của mảng
14     }
15
16     cout << "output = ";
17     // Duyệt qua tất cả các dãy con của mảng A
18     for(int i = 0; i < n; i++){
19         for(int j = i; j < n; j++){
20             for(int k = i; k <= j; k++){
21                 cout << A[k] << " ";
22             }
23             cout << endl;
24         }
25     }
26     delete [] A; // Giải phóng bộ nhớ đã cấp phát cho mảng A
27     return 0;
28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833
```

Hình 16. Bài 1.9_LamNH_0517

Kết quả chạy test cases trên Visual Studio Code:

Test case 1:

```
25     }
26     delete [] A; // Giải phóng bộ nhớ đã cấp phát cho mảng A
27     return 0;
28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Bai1.09_LamNH_0517
input = 2
3 2
output = 3
3 2
2

Hình 17. Bài 1.9_LamNH_0517 - TestCase 1

Test case 2:

```
25     }
26     delete [] A; // Giải phóng bộ nhớ đã cấp phát cho mảng A
27     return 0;
28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Bai1.09_LamNH_0517
input = 3
5 1 4
output = 5
5 1
5 1 4
1
1 4
4

Hình 18. Bài 1.9_LamNH_0517 - TestCase 2

Test case 3:

```
25     }
26     delete [] A; // Giải phóng bộ nhớ đã cấp phát cho mảng A
27     return 0;
28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Bai1.09_LamNH_0517
input = 10
1 4 -2 4 65 8 -9 12 5 67
output = 1
1 4
1 4 -2
1 4 -2 4
1 4 -2 4 65
1 4 -2 4 65 8
1 4 -2 4 65 8 -9
1 4 -2 4 65 8 -9 12
1 4 -2 4 65 8 -9 12 5
1 4 -2 4 65 8 -9 12 5 67
4
4 -2
4 -2 4
4 -2 4 65
4 -2 4 65 8
4 -2 4 65 8 -9
4 -2 4 65 8 -9 12
4 -2 4 65 8 -9 12 5
4 -2 4 65 8 -9 12 5 67
-2
-2 4
-2 4 65
-2 4 65 8
-2 4 65 8 -9
-2 4 65 8 -9 12

Hình 19. Bài 1.9_LamNH_0517 - TestCase 3 - 1

```

28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833

```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
<pre> -2 4 65 8 -9 12 5 -2 4 65 8 -9 12 5 67 4 4 65 4 65 8 4 65 8 -9 4 65 8 -9 12 4 65 8 -9 12 5 4 65 8 -9 12 5 67 65 65 8 65 8 -9 65 8 -9 12 65 8 -9 12 5 65 8 -9 12 5 67 8 8 -9 8 -9 12 8 -9 12 5 8 -9 12 5 67 -9 -9 12 -9 12 5 -9 12 5 67 12 12 5 12 5 67 5 5 67 67 cd "c:\Users\user\OneDrive - Hanoi University of Science and </pre>					

Hình 20. Bài 1.9_LamNH_0517 - TestCase 3 - 2

Test case 4:

```

28 }
29 // Nguyễn Hoàng Lâm - 0517 - 732833

```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
<pre> Bai1.09_LamNH_0517 input = 20 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39 output = 92 92 94 92 94 85 92 94 85 15 92 94 85 15 57 92 94 85 15 57 12 92 94 85 15 57 12 18 92 94 85 15 57 12 18 19 92 94 85 15 57 12 18 19 18 92 94 85 15 57 12 18 19 18 45 92 94 85 15 57 12 18 19 18 45 49 92 94 85 15 57 12 18 19 18 45 49 55 92 94 85 15 57 12 18 19 18 45 49 55 28 92 94 85 15 57 12 18 19 18 45 49 55 28 36 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39 94 94 85 94 85 15 94 85 15 57 94 85 15 57 12 94 85 15 57 12 18 94 85 15 57 12 18 19 94 85 15 57 12 18 19 18 94 85 15 57 12 18 19 18 45 94 85 15 57 12 18 19 18 45 49 94 85 15 57 12 18 19 18 45 49 55 94 85 15 57 12 18 19 18 45 49 55 28 94 85 15 57 12 18 19 18 45 49 55 28 36 94 85 15 57 12 18 19 18 45 49 55 28 36 16 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39 </pre>					

Hình 21. Bài 1.9_LamNH_0517 - TestCase 4 - 1

```
85
85 15
85 15 57
85 15 57 12
85 15 57 12 18
85 15 57 12 18 19
85 15 57 12 18 19 18
85 15 57 12 18 19 18 45
85 15 57 12 18 19 18 45 49
85 15 57 12 18 19 18 45 49 55
85 15 57 12 18 19 18 45 49 55 28
85 15 57 12 18 19 18 45 49 55 28 36
85 15 57 12 18 19 18 45 49 55 28 36 16
85 15 57 12 18 19 18 45 49 55 28 36 16 100
85 15 57 12 18 19 18 45 49 55 28 36 16 100 90
85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70
85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39
85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39
15
15 57
15 57 12
15 57 12 18
15 57 12 18 19
15 57 12 18 19 18
15 57 12 18 19 18 45
15 57 12 18 19 18 45 49
15 57 12 18 19 18 45 49 55
15 57 12 18 19 18 45 49 55 28
15 57 12 18 19 18 45 49 55 28 36
15 57 12 18 19 18 45 49 55 28 36 16
15 57 12 18 19 18 45 49 55 28 36 16 100
15 57 12 18 19 18 45 49 55 28 36 16 100 90
15 57 12 18 19 18 45 49 55 28 36 16 100 90 70
15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39
15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39
57
57 12
57 12 18
57 12 18 19
57 12 18 19 18
57 12 18 19 18 45
57 12 18 19 18 45 49
57 12 18 19 18 45 49 55
57 12 18 19 18 45 49 55 28
57 12 18 19 18 45 49 55 28 36
57 12 18 19 18 45 49 55 28 36 16
57 12 18 19 18 45 49 55 28 36 16 100
57 12 18 19 18 45 49 55 28 36 16 100 90
57 12 18 19 18 45 49 55 28 36 16 100 90 70
57 12 18 19 18 45 49 55 28 36 16 100 90 70 39
```

Hình 22. Bài 1.9_LamNH_0517 - TestCase 4 - 2

```
57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39
12
12 18
12 18 19
12 18 19 18
12 18 19 18 45
12 18 19 18 45 49
12 18 19 18 45 49 55
12 18 19 18 45 49 55 28
12 18 19 18 45 49 55 28 36
12 18 19 18 45 49 55 28 36 16
12 18 19 18 45 49 55 28 36 16 100
12 18 19 18 45 49 55 28 36 16 100 90
12 18 19 18 45 49 55 28 36 16 100 90 70
12 18 19 18 45 49 55 28 36 16 100 90 70 39
12 18 19 18 45 49 55 28 36 16 100 90 70 39 39
18
18 19
18 19 18
18 19 18 45
18 19 18 45 49
18 19 18 45 49 55
18 19 18 45 49 55 28
18 19 18 45 49 55 28 36
18 19 18 45 49 55 28 36 16
18 19 18 45 49 55 28 36 16 100
18 19 18 45 49 55 28 36 16 100 90
18 19 18 45 49 55 28 36 16 100 90 70
18 19 18 45 49 55 28 36 16 100 90 70 39
18 19 18 45 49 55 28 36 16 100 90 70 39 39
19
19 18
19 18 45
19 18 45 49
19 18 45 49 55
19 18 45 49 55 28
19 18 45 49 55 28 36
19 18 45 49 55 28 36 16
19 18 45 49 55 28 36 16 100
19 18 45 49 55 28 36 16 100 90
19 18 45 49 55 28 36 16 100 90 70
19 18 45 49 55 28 36 16 100 90 70 39
19 18 45 49 55 28 36 16 100 90 70 39 39
18
18 45
18 45 49
18 45 49 55
18 45 49 55 28
18 45 49 55 28 36
18 45 49 55 28 36 16
```

Hình 23. Bài 1.9_LamNH_0517 - TestCase 4 - 3

```
18 45 49 55 28 36 16 100
18 45 49 55 28 36 16 100 90
18 45 49 55 28 36 16 100 90 70
18 45 49 55 28 36 16 100 90 70 39
18 45 49 55 28 36 16 100 90 70 39 39
45
45 49
45 49 55
45 49 55 28
45 49 55 28 36
45 49 55 28 36 16
45 49 55 28 36 16 100
45 49 55 28 36 16 100 90
45 49 55 28 36 16 100 90 70
45 49 55 28 36 16 100 90 70 39
45 49 55 28 36 16 100 90 70 39 39
49
49 55
49 55 28
49 55 28 36
49 55 28 36 16
49 55 28 36 16 100
49 55 28 36 16 100 90
49 55 28 36 16 100 90 70
49 55 28 36 16 100 90 70 39
49 55 28 36 16 100 90 70 39 39
55
55 28
55 28 36
55 28 36 16
55 28 36 16 100
55 28 36 16 100 90
55 28 36 16 100 90 70
55 28 36 16 100 90 70 39
55 28 36 16 100 90 70 39 39
28
28 36
28 36 16
28 36 16 100
28 36 16 100 90
28 36 16 100 90 70
28 36 16 100 90 70 39
28 36 16 100 90 70 39 39
36
36 16
36 16 100
36 16 100 90
36 16 100 90 70
36 16 100 90 70 39
36 16 100 90 70 39 39
```

Hình 24. Bài 1.9_LamNH_0517 - TestCase 4 - 4

```
16
16 100
16 100 90
16 100 90 70
16 100 90 70 39
16 100 90 70 39 39
100
100 90
100 90 70
100 90 70 39
100 90 70 39 39
90
90 70
90 70 39
90 70 39 39
70
70 39
70 39 39
39
39 39
39
```

Hình 25. Bài 1.9_LamNH_0517 - TestCase 4-5

Bài 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.

Input: Dòng đầu chứa n là kích thước của mảng.

n dòng tiếp theo mô tả ma trận đầu tiên.

n dòng cuối cùng mô tả ma trận thứ hai.

Output: Với mỗi testcase, in ra tổng và tích của 2 ma trận.

Ví dụ:

Input:

2

2 1

5 3

-1 4

3 6

Output:

1 5

8 9

1 14

4 38

Bài làm

Chương trình (Trên Visual Code Studio):

```
1  /* Nguyễn Hoàng Lâm - 0517 - 732833
2  |   Bài 1.9. (BTVN) Viết chương trình nhập vào 2 ma trận vuông cùng kích thước  $n \times n$ , trong đó
3  |    $n$  nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình.*/
4  #include <stdio.h>
5  // Hàm cấp phát bộ nhớ động cho ma trận kích thước  $n \times n$  và gán con trỏ đến ma trận này vào biến A.
6  void ALLOCATE_MEMORY(int ***A, int n){
7  |   *A = new int*[n]; // Cấp phát mảng con trỏ chứa n hàng
8  |   for(int i = 0; i < n; i++){
9  |       (*A)[i] = new int[n]; // Cấp phát mảng số nguyên có n cột cho từng hàng của ma trận
10 |   }
11 |   }
12 // Hàm nhập giá trị cho các phần tử của ma trận có kích thước  $n \times n$ .
13 void INPUT(int **A, int n){
14 |   for(int i = 0; i < n; i++){
15 |       for(int j = 0; j < n; j++){
16 |           scanf("%d", &A[i][j]); // Nhập giá trị cho từng phần tử của ma trận
17 |       }
18 |   }
19 |   }
20 // Hàm thực hiện phép cộng hai ma trận A và B có kích thước  $n \times n$ , in ra kết quả.
21 void SUM(int **A, int **B, int n){
22 |   for(int i = 0; i < n; i++){
23 |       for(int j = 0; j < n; j++){
24 |           printf("%d ", A[i][j] + B[i][j]); // In giá trị của từng phần tử trong ma trận tổng
25 |       }
26 |       printf("\n"); // Xuống dòng sau mỗi hàng của ma trận
27 |   }
28 |   }
```

Hình 26. Bài 1.10_LamNH_0517 - Code 1

```

29 // Hàm thực hiện phép nhân hai ma trận A và B có kích thước nxn, in ra kết quả.
30 void PRODUCT(int **A, int **B, int n){
31     for(int i = 0; i<n; i++){
32         for(int j = 0; j<n; j++){
33             int Pro = 0;
34             for(int k = 0; k<n; k++){
35                 Pro += A[i][k]*B[k][j]; // Quy tắc nhân hai ma trận
36             }
37             printf("%d ", Pro); // In giá trị của từng phần tử trong ma trận tích
38         }
39         printf("\n"); // Xuống dòng sau mỗi hàng của ma trận
40     }
41 }
42 // Hàm giải phóng bộ nhớ đã cấp phát cho ma trận.
43 void FREE_MEMORY(int **A, int n){
44     for(int i = 0; i<n; i++){
45         delete[] A[i]; // Giải phóng bộ nhớ cho từng hàng của ma trận
46     }
47     delete[] A; // Giải phóng bộ nhớ cho mảng con trỏ
48 }

```

Hình 27. Bài 1.10_LamNH_0517 - Code 2

```

49 int main(){
50     int n, **A, **B;
51     printf("Bai.10_LamNH_0517 \ninput = \n");
52     scanf("%d", &n);
53     ALLOCATE_MEMORY(&A, n); // Cấp phát bộ nhớ cho ma trận A
54     INPUT(A, n); // Nhập giá trị cho ma trận A
55     ALLOCATE_MEMORY(&B, n); // Cấp phát bộ nhớ cho ma trận B
56     INPUT(B, n); // Nhập giá trị cho ma trận B
57
58     printf("output = \n");
59     SUM(A, B, n); // Thực hiện phép cộng và in ra kết quả
60     PRODUCT(A, B, n); // Thực hiện phép nhân và in ra kết quả
61
62     FREE_MEMORY(B, n); // Giải phóng bộ nhớ của ma trận B
63     FREE_MEMORY(A, n); // Giải phóng bộ nhớ của ma trận A
64     return 0;
65 }
66 // Nguyễn Hoàng Lâm - 0517 - 732833

```

Hình 28. Bài 1.10_LamNH_0517 - Code - 3

Kết quả chạy test cases trên Visual Studio Code:

Test case 1:

```

62     FREE_MEMORY(B, n); // Giải phóng bộ nhớ của ma trận B
63     FREE_MEMORY(A, n); // Giải phóng bộ nhớ của ma trận A
64     return 0;
65 }
66 // Nguyễn Hoàng Lâm - 0517 - 732833

```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
<p>Bai1.10_LamNH_0517</p> <p>input =</p> <p>2</p> <p>2 1</p> <p>5 3</p> <p>-1 4</p> <p>3 6</p> <p>output =</p> <p>1 5</p> <p>8 9</p> <p>1 14</p> <p>4 38</p>					

Hình 29. Bài 1.10_LamNH_0517 - TestCase 1

Test Case 2:

```

62     FREE_MEMORY(B, n); // Giải phóng bộ nhớ của ma trận B
63     FREE_MEMORY(A, n); // Giải phóng bộ nhớ của ma trận A
64     return 0;
65 }
66 // Nguyễn Hoàng Lâm - 0517 - 732833

```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
<p>Bai1.10_LamNH_0517</p> <p>input =</p> <p>3</p> <p>5 1 7</p> <p>1 2 3</p> <p>4 9 8</p> <p>3 -1 8</p> <p>2 5 7</p> <p>4 9 6</p> <p>output =</p> <p>8 0 15</p> <p>3 7 10</p> <p>8 18 14</p> <p>45 63 89</p> <p>19 36 40</p> <p>62 113 143</p>					

Hình 30. Bài 1.10_LamNH_0517 - TestCase 2

Test Case 3:

```

Buoi1_Pointer_DynamicAllocation > Bai1-10_LamNH_0517.cpp
1  /* Nguyễn Hoàng Lâm - 0517 - 732833
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Bai1.10_LamNH_0517
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

```

Hình 31. Bài 1.10_LamNH_0517 - TestCase 3

Test Case 4:

```

1  /* Nguyễn Hoàng Lâm - 0517 - 732833
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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
output =
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
103 149 110 112 54 96 110 49 96 120
78 110 38 90 111 53 41 134 90 132
105 116 153 79 101 141 112 92 94 122
21382 29723 24124 20954 26397 29918 27265 25457 28350 32407
21683 29432 22452 24803 29076 34127 27543 26666 31160 31948
16399 21213 21643 22126 20399 24994 24038 20295 22911 24376
21480 30091 22331 21421 25670 34932 27251 23402 29239 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

```

Hình 32. Bài 1.10_LamNH_0517 - TestCase 4