

Contents

Bài thực hành số 1 – Tuần 32.....	2
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.....	2
Bài tập 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 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.	7
Bài tập 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???	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.	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.	17
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.....	21
Bài tập 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.	27
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.	30

Nguyễn Văn Duy – 20215334
Bài thực hành số 1 – Tuần 32

Bài tập 1.1. Viết một chương trình C nhập vào 3 số nguyên. Thiết lập một con trỏ để lần lượt trở tới 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 trở tới 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:

Dữ liệu đầu vào	Kết quả
3 4 5	Enter three integers: The three integers are: x = 3 y = 4 z = 5

Đáp án: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Van Duy - 20215334
2 /*
3 Bài 1.1. Viết một chương trình C nhập vào 3 số nguyên.
4 Thiết lập một con trỏ để lần lượt trở tới từng số nguyên
5 và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.
6 */
7 #include <stdio.h>
8 int main(){
9     int x, y, z;
10    int* ptr;
11    printf("Enter three integers: ");
12    scanf("%d %d %d", &x, &y, &z);
13    // Nhập 3 số nguyên x, y, z
14
15    printf("\nThe three integers are:\n");
16    ptr = &x; // Gán địa chỉ của x cho ptr
17    printf("x = %d\n", *ptr);
18
19    //*****
20    # YOUR CODE HERE #
21    //*****
22    ptr = &y; // Gán địa chỉ của y cho ptr
23
24    printf("y = %d\n", *ptr);
25    ptr = &z; // Gán địa chỉ của z cho ptr
26    printf("z = %d\n", *ptr);
27
28    return 0;
29 }
30 // Nguyen Van Duy - 20215334
```

Chấm thử

Chấm điểm

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
✓	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	✓

Hoàn thành được tất cả các bộ mẫu ✓

Trang tiếp

// Nguyen Van Duy - 20215334

/*

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.

*/

include <stdio.h>

int main(){

int x, y, z;

int* ptr;

printf("Enter three integers: ");

scanf("%d %d %d", &x, &y, &z);

// Nhập 3 số nguyên x, y, z

printf("\nThe three integers are:\n");

ptr = &x; // Gán địa chỉ của x cho ptr

printf("x = %d\n", *ptr);

/******

YOUR CODE HERE

*****/

ptr = &y; // Gán địa chỉ của y cho ptr

printf("y = %d\n", *ptr);

ptr = &z; // Gán địa chỉ của z cho ptr

printf("z = %d\n", *ptr);

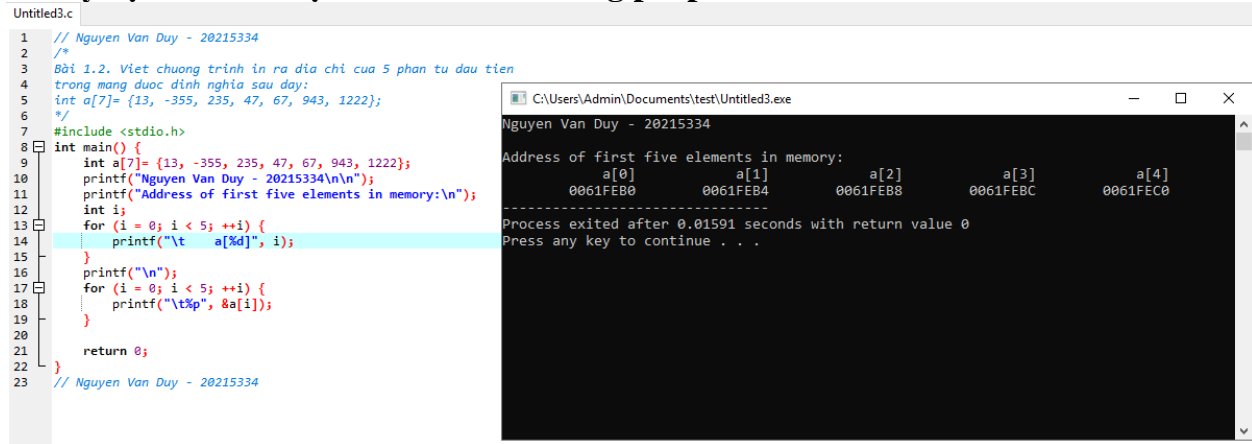
```
    return 0;  
}  
// Nguyen Van Duy - 20215334
```

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 &

The image shows a C program in a text editor and its execution in a command prompt. The program defines an array 'a' with 7 integers and prints the address of the first five elements. The output shows the memory addresses for a[0] through a[4].

```
1 // Nguyen Van Duy - 20215334
2 /*
3  Bài 1.2. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên
4  trong mảng được định nghĩa sau đây:
5  int a[7]= {13, -355, 235, 47, 67, 943, 1222};
6  */
7 #include <stdio.h>
8 int main() {
9     int a[7]= {13, -355, 235, 47, 67, 943, 1222};
10    printf("Nguyen Van Duy - 20215334\n\n");
11    printf("Address of first five elements in memory:\n");
12    int i;
13    for (i = 0; i < 5; ++i) {
14        printf("\t a[%d]", i);
15    }
16    printf("\n");
17    for (i = 0; i < 5; ++i) {
18        printf("\t%p", &a[i]);
19    }
20
21    return 0;
22 }
23 // Nguyen Van Duy - 20215334
```

```
Nguyen Van Duy - 20215334
Address of first five elements in memory:
a[0]      a[1]      a[2]      a[3]      a[4]
0061FEB0  0061FEB4  0061FEB8  0061FEBc  0061FEC0
-----
Process exited after 0.01591 seconds with return value 0
Press any key to continue . . .
```

```
// Nguyen Van Duy - 20215334
```

```
/*
```

Bài 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};
```

```
*/
```

```
#include <stdio.h>
```

```
int main() {
```

```
    int a[7]= {13, -355, 235, 47, 67, 943, 1222};
```

```
    printf("Nguyen Van Duy - 20215334\n\n");
```

```
    printf("Address of first five elements in memory:\n");
```

```
    int i;
```

```
    for (i = 0; i < 5; ++i) {
```

```
        printf("\t a[%d]", i);
```

```
    }
```

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

```
for (i = 0; i < 5; ++i) {  
    printf("\t%p", &a[i]);  
}  
  
return 0;  
}  
// Nguyen Van Duy – 20215334
```

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:

Dữ liệu đầu vào	Kết quả
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

Đáp án: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Van Duy - 20215334
2 /*
3  Bài 1.3. Viết chương trình yêu cầu nhập giá trị cho
4  3 biến số nguyên x, y, z kiểu int.
5  Sau đó sử dụng duy nhất một con trỏ để cộng giá trị
6  của mỗi biến thêm 100.
7  */
8 #include <stdio.h>
9 int main() {
10     int x, y, z;
11     int *ptr;
12     scanf("%d %d %d", &x, &y, &z);
13     // Nhập 3 số nguyên 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     *****/
20     ptr = &x; // Gán địa chỉ của x cho ptr
21     *ptr += 100; // Cộng thêm 100
22     ptr = &y; // Gán địa chỉ của y cho ptr
23
24     *ptr += 100; // Cộng thêm 100
25     ptr = &z; // Gán địa chỉ của z cho ptr
26     *ptr += 100; // Cộng thêm 100
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 }
```

Chấm thử

Chấm điểm

Nguyễn Văn Duy – 20215334

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
✓	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	✓

Hoàn thành được tất cả các bộ mẫu ✓

Trang tiếp

// Nguyen Van Duy - 20215334

/*

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);

// Nhập 3 số nguyên 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

*****/

ptr = &x; // Gán địa chỉ của x cho ptr


```
*ptr += 100; // Cong them 100
ptr = &y; // Gan dia chi cua y cho ptr
*ptr += 100; // Cong them 100
ptr = &z; // Gan dia chi cua z cho ptr
*ptr += 100; // Cong them 100

printf("Once again, here are the values of x, y, and z:\n");
printf("%d %d %d\n", x, y, z);
return 0;
}
// Nguyen Van Duy - 20215334
```

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

Đáp án: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Van Duy - 20215334
2 /*
3 Bài 1.4. Viết hàm countEven(int*, int)
4 nhận một mảng số nguyên và kích thước của mảng,
5 trả về số lượng số chẵn trong mảng???
6 */
7 #include <iostream>
8
9 using namespace std;
10
11 int counteven(int* arr, int size){
12     int count = 0;
13
14     /*
15     # YOUR CODE HERE #
16     */
17     for (int i = 0; i < size; ++i) {
18         if (!(arr[i] & 1)) {
19             ++count;
20         }
21     }
22
23     return count;
24 }
25 // Nguyen Van Duy - 20215334

```

Chấm thử

Chấm điểm

	Test	Kết quả đúng	Kết quả chương trình	
✓	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	✓

Hoàn thành được tất cả các bộ mẫu ✓

Trang tiếp

// Nguyen Van Duy - 20215334

/*

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???

*/

#include <iostream>

```
using namespace std;
```

```
int counteven(int* arr, int size){
```

```
    int count = 0;
```

```
    /*****
```

```
    # YOUR CODE HERE #
```

```
    *****/
```

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

```
        if (!(arr[i] & 1)) {
```

```
            ++count;
```

```
        }
```

```
    }
```

```
    return count;
```

```
}
```

```
// Nguyen Van Duy - 20215334
```

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.

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.

```
double* maximum(double* a, int size){
    double *max;
    max = a;
    if (a==NULL) return NULL;

    /*****
    # YOUR CODE HERE #
    *****/

    return max;
}
```

For example:

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

Đáp án: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Van Duy - 20215334
2 /*
3  Bài 1.5. Viết hàm trả về con trỏ trỏ tới
4  giá trị lớn nhất của một mảng các số double.
5  Nếu mảng rỗng hãy trả về NULL.
6  */
7 #include <stdio.h>
8
9 double* maximum(double* a, int size){
10     double *max;
11     max = a;
12     if (a==NULL) return NULL;
13
14     /*****
15     # YOUR CODE HERE #
16     *****/
17     for (int i = 0; i < size; ++i) {
18         if (*(a+i) > *max) {
19             max = a+i;
20         }
21     }
22
23     return max;
24 }
25 // Nguyen Van Duy - 20215334
```

Chấm thử

Chấm điểm

	Test	Kết quả đúng	Kết quả chương trình	
✓	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	✓

Hoàn thành được tất cả các bộ mẫu ✓

```
// Nguyen Van Duy - 20215334
```

```
/*
```

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.

```
*/
```

```
#include <stdio.h>
```

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

```
    double *max;
```

```
    max = a;
```

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

```
    /*****
```

```
    # YOUR CODE HERE #
```

```
    *****/
```

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

```
        if (*(a+i) > *max) {
```

```
            max = a+i;
```

```
        }
```

```
    }
```

```
    return max;
```

```
}
```

```
// Nguyen Van Duy - 20215334
```

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	Kết quả
<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>	<pre>5 2 6 5 3 9 9 5 -1 4</pre>

Đáp án: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Van Duy - 20215334
2 /*
3  Bài 1.6. Viết hàm đảo ngược một mảng
4  các số nguyên theo hai cách:
5  dùng chỉ số và dùng con trỏ.
6  */
7 #include <iostream>
8
9 using namespace std;
10
11 void reversearray(int arr[], int size){
12     int l = 0, r = size - 1, tmp;
13
14     /*****
15     # YOUR CODE HERE #
16     *****/
17     while (l < r) {
18         // swap
19         tmp = arr[l];
20         arr[l] = arr[r];
21         arr[r] = tmp;
22         ++l;
23         --r;
24     }
25 }
26
27 void ptr_reversearray(int *arr, int size){
28     int l = 0, r = size - 1, tmp;
29
30     /*****
31     # YOUR CODE HERE #
32     *****/
33     while (l < r) {
34         // swap
35         tmp = *(arr+l);
36         *(arr+l) = *(arr+r);
37         *(arr+r) = tmp;
38         ++l;
39         --r;
40     }
41 }
```

```
39 |     --r;
40 | }
41 | }
42 | // Nguyen Van Duy - 20215334
```

Chấm thử

Chấm điểm

	Test	Kết quả đúng	Kết quả chương trình	
✓	<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	✓

Hoàn thành được tất cả các bộ mẫu ✓

Trang tiếp

// Nguyen Van Duy - 20215334

/*

Bài 1.6. Viết hàm đảo ngược một mảng

các số nguyên theo hai cách:

dùng chỉ số và dùng con trỏ.

*/

#include <iostream>

using namespace std;

void reversearray(int arr[], int size){

int l = 0, r = size - 1, tmp;

/******

YOUR CODE HERE

*****/

```
while (l < r) {  
    // swap  
    tmp = arr[l];  
    arr[l] = arr[r];  
    arr[r] = tmp;  
    ++l;  
    --r;  
}  
}
```

```
void ptr_reversearray(int *arr, int size){
```

```
    int l = 0, r = size - 1, tmp;
```

```
    /*****
```

```
    # YOUR CODE HERE #
```

```
    *****/
```

```
    while (l < r) {
```

```
        // swap
```

```
        tmp = *(arr+l);
```

```
        *(arr+l) = *(arr+r);
```

```
        *(arr+r) = tmp;
```

```
        ++l;
```

```
        --r;
```

```
    }
```

```
}
```

```
// Nguyen Van Duy – 20215334
```


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:

Dữ liệu đầu vào	Kết quả
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

Đáp án: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Van Duy - 20215334
2 /*
3  Bài 1.7. Viết chương trình nhập vào một mảng các
4  số nguyên với số lượng các phần tử nhập từ bàn phím.
5  Sau đó sắp xếp mảng theo thứ tự tăng dần.
6  Hiển thị danh sách mảng trước và sau khi sắp xếp.
7  */
8  #include <stdio.h>
9  #include <stdlib.h>
10 int *a;
11 int n, tmp;
12
13 int main(){
14     printf("Enter the number of elements: ");
15     scanf("%d", &n);
16
17     //Allocate memory
18
19     /*****
20     # YOUR CODE HERE #
21     *****/
22     a = (int *) malloc(n * sizeof(int));

```

```

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

```

```

44                 *(a+j+1) = tmp;
45             }
46         }
47     }
48
49     printf("The sorted array is: \n");
50     for(int i = 0; i < n; i++)
51         printf("%d ", *(a + i));
52     printf("\n");
53
54     free(a);
55     return 0;
56 }
57 // Nguyen Van Duy - 20215334

```

Chăm thử

Chăm điểm

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
✓	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	✓

Hoàn thành được tất cả các bộ mẫu ✓

Trang tiếp

// Nguyen Van Duy - 20215334

/*

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.

*/

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int *a;
```

```
int n, tmp;
```

```
int main(){
```

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

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

```
    //Allocate memory
```

```
    /*****
```

```
# YOUR CODE HERE #

*****/

a = (int *) malloc(n * sizeof(int));

/*****/

for(int i = 0; i < n; i++)
    scanf("%d", a + i);

printf("The input array is: \n");
for(int i = 0; i < n; i++)
    printf("%d ", *(a + i));
printf("\n");

//#Sort array

/*****

# YOUR CODE HERE #

*****/

for (int i = 0; i < n-1; ++i) {
    for (int j = 0; j < n-i-1; ++j) {
        if (*(a+j) > *(a+j+1)) {
            // swap
            int tmp = *(a+j);
            *(a+j) = *(a+j+1);
            *(a+j+1) = tmp;
        }
    }
}
```

```
}
```

```
printf("The sorted array is: \n");
```

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

```
    printf("%d ", *(a + i));
```

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

```
free(a);
```

```
return 0;
```

```
}
```

```
// Nguyen Van Duy - 20215334
```

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_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 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 theo dạng địa chỉ, ví dụ sử dụng con trỏ cấp 3 dạng `int ***mt`.

For example:

Dữ liệu đầu vào	Kết quả
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

Đáp án: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Van Duy - 20215334
2 /*
3 Bài 1.8. Viết chương trình nhập vào một ma trận
4 2 chiều kích thước m*n với m và n nhập từ bàn phím.
5 Sau đó đưa ra tổng các phần tử chẵn của ma trận đó.
6 */
7 #include <stdio.h>
8 #include <stdlib.h>
9 void allocate_mem(int ***mt, int m, int n){
10     //Allocate memory for the matrix
11
12     /*
13     # YOUR CODE HERE #
14     */
15     *mt = (int **) malloc(m * sizeof(int *));
16     int i;
17     for (i = 0; i < m; ++i) {
18         *(mt+i) = (int *) malloc(n * sizeof(int));
19     }
20 }
21
22

```

```

23 void input(int **mt, int m, int n){
24     //Input elements of the matrix
25
26     /*
27     # YOUR CODE HERE #
28     */
29     int i, j;
30     for (i = 0; i < m; ++i) {
31         for (j = 0; j < n; ++j) {
32             printf("mt[%d][%d] = ", i, j);
33             scanf("%d", *(mt + i) + j);
34         }
35     }
36 }
37
38 void output(int **mt, int m, int n){
39     //Print all elements of the matrix
40
41     /*
42     # YOUR CODE HERE #
43     */

```

```

44     int i, j;
45     for (i = 0; i < m; ++i) {
46         for (j = 0; j < n; ++j) {
47             printf("%d ", (*(mt + i) + j));
48         }
49         printf("\n");
50     }
51 }
52
53 int process(int **mt, int m, int n){
54     int tong = 0;
55     /* Calculate the sum of all even elements in the matrix
56
57     *****
58     # YOUR CODE HERE #
59     *****/
60     int i, j;
61     for (i = 0; i < m; ++i) {
62         for (j = 0; j < n; ++j) {
63             if (!(*(mt + i) + j) & 1) {
64                 tong += (*(mt + i) + j);
65             }
66         }
67     }
68
69     return tong;
70 }
71

```

```

72 void free_mem(int **mt, int m, int n){
73     /* Free memory
74
75     *****
76     # YOUR CODE HERE #
77     *****/
78     int i;
79     for (i = 0; i < m; ++i) {
80         free(mt[i]);
81     }
82     free(mt);
83 }
84
85 int main(){
86     int m, n, **mt;
87     printf("Enter m, n = ");
88     scanf("%d%d", &m, &n);
89     allocate_mem(&mt, m, n);
90     input(mt, m, n);
91     output(mt, m, n);
92     printf("The sum of all even elements is %d", process(mt, m, n));
93     free_mem(mt, m, n);
94     return 0;
95 }
96 // Nguyen Van Duy - 20215334

```

Chấm thử

Chấm điểm

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
✓	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] = 1 2 4 5 The sum of all even elements is 6	✓

Hoàn thành được tất cả các bộ mẫu ✓

Kết thúc bài thi ...

// Nguyen Van Duy - 20215334

/*

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.

*/

#include <stdio.h>

#include <stdlib.h>

void allocate_mem(int ***mt, int m, int n){

 //#Allocate memory for the matrix

 /*****

 # YOUR CODE HERE #

 *****/

 *mt = (int **) malloc(m * sizeof(int *));

 int i;

 for (i = 0; i < m; ++i) {

 >(*mt+i) = (int *) malloc(n * sizeof(int));

 }

}

void input(int **mt, int m, int n){

 //#Input elements of the matrix

 /*****

 # YOUR CODE HERE #

```
*****/
int i, j;
for (i = 0; i < m; ++i) {
    for (j = 0; j < n; ++j) {
        printf("mt[%d][%d] = ", i, j);
        scanf("%d", *(mt + i) + j);
    }
}
}
```

```
void output(int **mt, int m, int n){
    // # Print all elements of the matrix
```

```

/*****
# YOUR CODE HERE #
*****/
int i, j;
for (i = 0; i < m; ++i) {
    for (j = 0; j < n; ++j) {
        printf("%d ", (*(mt + i) + j));
    }
    printf("\n");
}
}
```

```
int process(int **mt, int m, int n){
    int tong = 0;
```


//# Calculate the sum of all even elements in the matrix

/******

YOUR CODE HERE

*****/

int i, j;

for (i = 0; i < m; ++i) {

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

if (!(*(mt + i) + j) & 1) {

tong += *(mt + i) + j;

}

}

}

return tong;

}

void free_mem(int **mt, int m, int n){

//# Free memory

/******

YOUR CODE HERE

*****/

int i;

for (i = 0; i < m; ++i) {

free(mt[i]);

}

```
    free(mt);
}

int main(){
    int m, n, **mt;
    printf("Enter m, n = ");
    scanf("%d%d", &m, &n);
    allocate_mem(&mt, m, n);
    input(mt, m, n);
    output(mt, m, n);
    printf("The sum of all even elements is %d", process(mt, m, n));
    free_mem(mt, m, n);
    return 0;
}

// Nguyen Van Duy - 20215334
```

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

```

1 // Nguyen Van Duy - 20215334
2 /*
3  * Bài 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.
4  */
5 #include <iostream>
6
7 using namespace std;
8
9 void process(int *a, int n) {
10     for (int i = 0; i < n; ++i) {
11         string s;
12         for (int j = i; j < n; ++j) {
13             s += to_string(a[j]) + " ";
14             cout << s << endl;
15         }
16     }
17 }
18
19 int main() {
20     int *a, n;
21
22     // nhập giá trị cho n
23     cin >> n;
24
25     // Allocate memory for the array
26     a = new int[n];
27
28     // nhập giá trị cho mảng
29     for (int i = 0; i < n; ++i) {
30         cin >> a[i];
31     }
32
33     process(a, n);
34
35     // free memory
36     delete[] a;
37 }
38 // Nguyen Van Duy - 20215334
  
```

Run untitled16

```

C:\Users\Admin\CLionProjects\untitled16\cmake-build-debug\untitled16.exe
3
5 1 4
5
5 1
5 1 4
1
1 4
4
Process finished with exit code 0
  
```

```
// Nguyen Van Duy - 20215334

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

#include <iostream>

using namespace std;

void process(int *a, int n) {
    for (int i = 0; i < n; ++i) {
        string s;
        for (int j = i; j < n; ++j) {
            s += to_string(a[j]) + " ";
            cout << s << endl;
        }
    }
}

int main() {
    int *a, n;

    // nhap gia tri cho n
    cin >> n;

    // Allocate memory for the array
    a = new int[n];
```

```
// nhập giá trị cho mảng
for (int i = 0; i < n; ++i) {
    cin >> a[i];
}

process(a, n);

// free memory
delete[] a;
}
// Nguyen Van Duy – 20215334
```

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.

```

CMakeLists.txt  main.cpp ×
1 // Nguyen Van Duy - 20215334
2 /*
3  * Bài 1.10. Viết chương trình nhập vào 2 ma trận vuông cùng kích thước  $n \times n$ ,
4  * trong đó  $n$  nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó
5  * và đưa kết quả ra màn hình.
6  * Yêu cầu sử dụng cấp phát động để cấp phát bộ nhớ cho các ma trận.
7  */
8 #include <iostream>
9
10 using namespace std;
11
12 void allocate(int ***mt, int n) {
13     // Allocate memory for the matrix
14     *mt = new int * [n];
15     for (int i = 0; i < n; ++i) {
16         (*mt)[i] = new int[n];
17     }
18 }
19
20 void input(int **mt, int n) {
21     // nhập giá trị cho ma trận
22     for (int i = 0; i < n; ++i) {
23         for (int j = 0; j < n; ++j) {
24             cin >> mt[i][j];
25         }
26     }
27 }
28
29 void sum(int **mtA, int **mtB, int n) {
30     for (int i = 0; i < n; ++i) {
31         for (int j = 0; j < n; ++j) {
32             cout << mtA[i][j] + mtB[i][j] << " ";
33         }
34         cout << endl;
35     }
36 }
37
38 void mul(int **mtA, int **mtB, int n) {
39     for (int i = 0; i < n; ++i) {
40         for (int j = 0; j < n; ++j) {
41             int val = 0;
42             for (int k = 0; k < n; ++k) {
43                 val += mtA[i][k] * mtB[k][j];
44             }
45             cout << val << " ";
46         }
47         cout << endl;
48     }
49 }

```

```

50
51 void free(int **mt, int n){
52     // Free memory
53     for (int i = 0; i < n; ++i) {
54         delete[] mt[i];
55     }
56     if (n > 0) {
57         delete[] mt;
58     }
59 }
60
61 int main() {
62     int **mtA, **mtB, n;
63
64     // nhập giá trị cho n
65     cin >> n;
66
67     // Allocate memory
68     allocate(mt: &mtA, n);
69     allocate(mt: &mtB, n);
70
71     input(mt: mtA, n);
72     input(mt: mtB, n);
73
74     // output
75     sum(mtA, mtB, n);
76     mul(mtA, mtB, n);
77
78     // free memory
79     free(mt: mtA, n);
80     free(mt: mtB, n);
81 }
82 // Nguyen Van Duy - 20215334

```

Run untitled16 x

3
5 1 7
1 2 3
4 9 8
3 -1 8
2 5 7
4 9 6
8 0 15
3 7 10
8 18 14
45 63 89
19 36 40
62 113 143

untitled16 > main.cpp

```
// Nguyen Van Duy - 20215334
```

```
/*
```

```
* 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.
```

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
void allocate(int ***mt, int n) {
```

```
    // Allocate memory for the matrix
```

```
    *mt = new int * [n];
```

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

```
        (*mt)[i] = new int[n];
```

```
    }
```

```
}
```

```
void input(int **mt, int n) {
```

```
    // nhập giá trị cho ma trận
```

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

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



```
        cin >> mt[i][j];  
    }  
}  
}
```

```
void sum(int **mtA, int **mtB, int n) {  
    for (int i = 0; i < n; ++i) {  
        for (int j = 0; j < n; ++j) {  
            cout << mtA[i][j] + mtB[i][j] << " ";  
        }  
        cout << endl;  
    }  
}
```

```
void mul(int **mtA, int **mtB, int n) {  
    for (int i = 0; i < n; ++i) {  
        for (int j = 0; j < n; ++j) {  
            int val = 0;  
            for (int k = 0; k < n; ++k) {  
                val += mtA[i][k] * mtB[k][j];  
            }  
            cout << val << " ";  
        }  
    }
```

```
        cout << endl;
    }
}
```

```
void free(int **mt, int n){
    // Free memory
    for (int i = 0; i < n; ++i) {
        delete[] mt[i];
    }
    if (n > 0) {
        delete[] mt;
    }
}
```

```
int main() {
    int **mtA, **mtB, n;

    // nhap gia tri cho n
    cin >> n;

    // Allocate memory
    allocate(&mtA, n);
    allocate(&mtB, n);
```

```
input(mtA, n);  
input(mtB, n);  
  
// output  
sum(mtA, mtB, n);  
mul(mtA, mtB, n);  
  
// free memory  
free(mtA, n);  
free(mtB, n);  
}  
// Nguyen Van Duy - 20215334
```