

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CNTT&TT

*****  *****



BÁO CÁO THỰC HÀNH
HỌC PHẦN: KỸ THUẬT LẬP TRÌNH

Bài 1 – Tuần 7

Sinh viên thực hiện: **Tạ Quang Phổ**
MSSV: **20215450**
Mã lớp: **IT3040 – 732830**

Giảng viên hướng dẫn: **ThS. Lê Thị Hoa**

___. Năm học 2023-2024 .__

Contents

Bài thực hành số 1 – Tuần 32	3
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.	3
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.....	6
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???	7
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.	8
Bài tập 1.6. Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ.....	10
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.	12
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.....	16
Bài tập 1.9. Viết chương trình in ra tất cả các dãy con của một dãy cho trước.	21
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

Tạ Quang Phổ - 20215450

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 trỏ tới từng số nguyên và hiển thị kết quả giá trị tham chiếu ngược của con trỏ.

```

1 #include <stdio.h>
2
3 int main(){
4     int x, y, z;
5     int* ptr;
6     printf("Enter three integers: ");
7     scanf("%d %d %d", &x, &y, &z);
8     printf("\nThe three integers are:\n");
9     ptr = &x;
10    printf("x = %d\n", *ptr);
11
12    //*****
13    /* Ta Quang Pho - 20215450 */
14    ptr = &y;                //gán địa chỉ của biến y cho con trỏ ptr
15    printf("y = %d\n", *ptr); //in ra giá trị ở nhớ được trỏ bởi con trỏ ptr
16    ptr = &z;                //gán địa chỉ của biến z cho con trỏ ptr
17    printf("z = %d\n", *ptr); //in ra giá trị ở nhớ được trỏ bởi con trỏ ptr
18    //*****
19
20    return 0;
21 }

```

	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! ✓

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

```
int main(){
```

```
    int x, y, z;
```

```
    int* ptr;
```

```
    printf("Enter three integers: ");
```

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

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phô - 20215450

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

ptr = &x;

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

//*****//

/* Ta Quang Pho - 20215450 */

ptr = &y;           //gán địa chỉ của biến y cho con trỏ ptr
printf("y = %d\n", *ptr);    //in ra giá trị ô nhớ được trỏ bởi con trỏ ptr

ptr = &z;           //gán địa chỉ của biến z cho con trỏ ptr
printf("z = %d\n", *ptr);    //in ra giá trị ô nhớ được trỏ bởi con trỏ ptr

//*****//

return 0;

}
```

Tạ Quang Phổ - 20215450

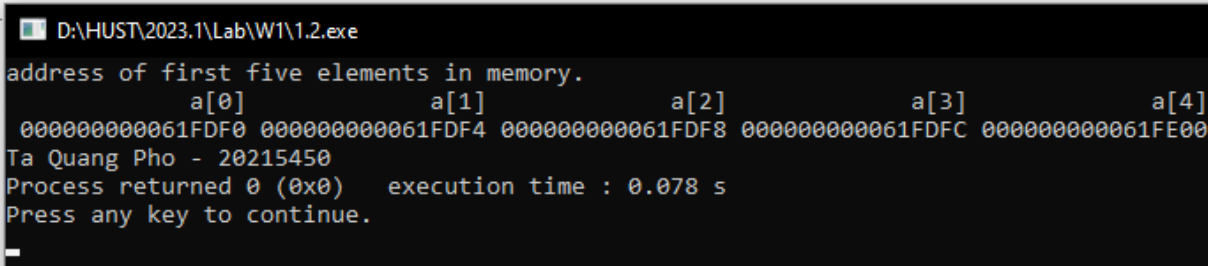
Bài tập 1.2. Viết chương trình in ra địa chỉ của 5 phần tử đầu tiên trong mảng được định nghĩa sau đây: `int a[7] = {13, -355, 235, 47, 67, 943, 1222};`

```
#include <stdio.h>

int main(){
    int a[7] = {13, -355, 235, 47, 67, 943, 1222};
    printf("address of first five elements in memory.\n");
    for (int i=0; i<5;i++) printf("          a[%d]", i);
    printf("\n");

    //*****//
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < 5; i++) {
        printf(" %p", a + i);
    }
    //*****//
    printf("\nTa Quang Pho - 20215450");

    return 0;
}
```



```
D:\HUST\2023.1\Lab\W1\1.2.exe
address of first five elements in memory.
          a[0]          a[1]          a[2]          a[3]          a[4]
000000000061FDF0 000000000061FDF4 000000000061FDF8 000000000061FD0C 000000000061FE00
Ta Quang Pho - 20215450
Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```

```
#include <stdio.h>

int main(){
    int a[7] = {13, -355, 235, 47, 67, 943, 1222};
    printf("address of first five elements in memory.\n");
    for (int i=0; i<5;i++) printf("          a[%d]", i);
    printf("\n");

    //*****//
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < 5; i++) {
        printf(" %p", a + i);
    }
    //*****//
    printf("\nTa Quang Pho - 20215450");

    return 0;
}
```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phổ - 20215450

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.

```

1 #include <stdio.h>
2 int main()
3 {
4     int x, y, z;
5     int *ptr;
6     scanf("%d %d %d", &x, &y, &z);
7     printf("Here are the values of x, y, and z:\n");
8     printf("%d %d %d\n", x, y, z);
9
10    //*****//
11    /* Ta Quang Pho - 20215450 */
12    ptr = &x;           //gán địa chỉ của biến x cho con trỏ ptr
13    *ptr += 100;         //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
14    ptr = &y;           //gán địa chỉ của biến y cho con trỏ ptr
15    *ptr += 100;         //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
16    ptr = &z;           //gán địa chỉ của biến z cho con trỏ ptr
17    *ptr += 100;         //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
18    //*****//
19
20    printf("Once again, here are the values of x, y, and z:\n");
21    printf("%d %d %d\n", x, y, z);
22    return 0;
23 }

```

	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! ✓

```

#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);

    //*****//
    /* Ta Quang Pho - 20215450 */
    ptr = &x;           //gán địa chỉ của biến x cho con trỏ ptr
    *ptr += 100;         //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
    ptr = &y;           //gán địa chỉ của biến y cho con trỏ ptr

```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phổ - 20215450

```
*ptr += 100;          //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
ptr = &z;             //gán địa chỉ của biến z cho con trỏ ptr
*ptr += 100;          //tăng giá trị của ô nhớ được trỏ bằng ptr thêm 100
//*****//
```

```
printf("Once again, here are the values of x, y, and z:\n");
printf("%d %d %d\n", x, y, z);
return 0;
```

```
}
```

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???

```
1 int counteven(int* arr, int size){
2     int count = 0;
3
4     //*****//
5     /* Ta Quang Pho - 20215450 */
6     for (int i = 0; i < size; i++) {          //vòng lặp size lần với biến chạy i, chạy từ 0 đến size - 1
7         if (arr[i] % 2 == 0) count++;        //kiểm tra nếu phần tử thứ i của mảng chia hết cho 2 thì
8                                             //tăng count thêm 1 đơn vị
9     }
10    //*****//
11
12    return count;                            // trả về giá trị của biến count
13 }
```

	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! ✓

```
int counteven(int* arr, int size){
    int count = 0;

    //*****//
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < size; i++) { //vòng lặp size lần với biến chạy i, chạy từ 0 đến size - 1
        if (arr[i] % 2 == 0) count++; //kiểm tra nếu phần tử thứ i của mảng chia hết cho 2 thì
                                    //tăng count thêm 1 đơn vị
    }
```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phổ - 20215450

```

}
//*****//

return count;           // trả về giá trị của biến count
}

```

Bài tập 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.

```

1 double* maximum(double* a, int size){
2     double *max;
3     max = a;
4     if (a==NULL) return NULL;
5
6     //*****//
7     /* Ta Quang Pho - 20215450 */
8     for (int i = 1; i < size; i++) {           //vòng lặp size lần với biến chạy i, chạy từ 0 đến size - 1
9         if (*max < a[i]) max = a + i;         //so sánh giá trị của ô nhớ được trỏ bởi max
10                                            //và giá trị phần tử thứ i của mảng a;
11                                            //nếu nhỏ hơn thì gán địa chỉ phần tử thứ i của a cho max
12     }
13     //*****//
14
15     return max;
16 }

```

	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! ✓

```

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

    //*****//
    /* Ta Quang Pho - 20215450 */
    for (int i = 1; i < size; i++) {           //vòng lặp size lần với biến chạy i, chạy từ 0 đến size
- 1

```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phô - 20215450

```
if (*max < a[i]) max = a + i;    //so sánh giá trị của ô nhớ được trỏ bởi max
                                //và giá trị phần tử thứ i của mảng a;
                                //nếu nhỏ hơn thì gán địa chỉ phần tử thứ i của a cho max
}
//*****//

return max;
}
```

Tạ Quang Phổ - 20215450

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

```

1 void reversearray(int arr[], int size){
2     int l = 0, r = size - 1, tmp;
3
4     //*****//
5     /* Ta Quang Pho - 20215450 */
6     while (l <= r) { //vòng lặp có 2 biến chạy là l(left) và r(right)
7                     //vòng lặp dừng lại khi biến chạy trái > biến chạy phải
8         tmp = arr[l]; //tạo 1 biến trung gian lưu trữ giá trị phần tử bên trái đang xét
9         arr[l] = arr[r]; //gán giá trị phần tử r trong mảng cho phần tử l
10        arr[r] = tmp; //gán giá trị trung gian cho phần tử r trong mảng
11        l++; r--; //tăng con chạy trái thêm 1 và giảm con chạy phải đi 1 đơn vị
12    }
13    //*****//
14 }
15
16 void ptr_reversearray(int *arr, int size){
17     int l = 0, r = size - 1, tmp;
18
19     //*****//
20     /* Ta Quang Pho - 20215450 */
21     while (l <= r) { //vòng lặp có 2 biến chạy là l(left) và r(right)
22                     //vòng lặp dừng lại khi biến chạy trái >= biến chạy phải
23         tmp = *(arr + l); //tạo 1 biến trung gian lưu trữ giá trị phần tử bên trái đang xét
24         *(arr + l) = *(arr + r); //gán giá trị phần tử r trong mảng cho phần tử l
25         *(arr + r) = tmp; //gán giá trị trung gian cho phần tử r trong mảng
26         l++; r--; //tăng con chạy trái thêm 1 và giảm con chạy phải đi 1 đơn vị
27     }
28    //*****//
29 }

```

	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! ✓

```

void reversearray(int arr[], int size){
    int l = 0, r = size - 1, tmp;

    //*****//
    /* Ta Quang Pho - 20215450 */
    while (l <= r) { //vòng lặp có 2 biến chạy là l(left) và r(right)
                    //vòng lặp dừng lại khi biến chạy trái > biến chạy phải

```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phổ - 20215450

```

    tmp = arr[l];           //tạo 1 biến trung gian lưu trữ giá trị phần tử bên trái đang
xét
    arr[l] = arr[r];        //gán giá trị phần tử r trong mảng cho phần tử l
    arr[r] = tmp;          //gán giá trị trung gian cho phần tử r trong mảng
    l++; r--;              //tăng con chạy trái thêm 1 và giảm con chạy phải đi 1 đơn vị
}
//*****//
}

```

```

void ptr_reversearray(int *arr, int size){
    int l = 0, r = size - 1, tmp;

    //*****//
    /* Ta Quang Pho - 20215450 */
    while (l <= r) {        //vòng lặp có 2 biến chạy là l(left) và r(right)
                            //vòng lặp dừng lại khi biến chạy trái >= biến chạy phải
        tmp = *(arr + l);   //tạo 1 biến trung gian lưu trữ giá trị phần tử bên trái đang
xét
        *(arr + l) = *(arr + r); //gán giá trị phần tử r trong mảng cho phần tử l
        *(arr + r) = tmp;      //gán giá trị trung gian cho phần tử r trong mảng
        l++; r--;              //tăng con chạy trái thêm 1 và giảm con chạy phải đi 1 đơn vị
    }
    //*****//
}

```

Tạ Quang Phổ - 20215450

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.

Tạ Quang Phổ - 20215450

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int *a;
5 int n, tmp;
6
7 int main(){
8     printf("Enter the number of elements: ");
9     scanf("%d", &n);
10
11     //Allocate memory
12
13     //*****
14     /* Ta Quang Phu - 20215450 */
15     a = (int*) malloc (n * sizeof(int));    //cấp phát bộ nhớ kích thước n phần tử kiểu nguyên
16                                           //và gán địa chỉ bộ nhớ vừa cấp phát cho con trỏ a
17     //*****
18
19     for(int i = 0; i < n; i++)
20         scanf("%d", a + i);
21
22     printf("The input array is: \n");
23     for(int i = 0; i < n; i++)
24         printf("%d ", *(a + i));
25     printf("\n");
26
27     //Sort array
28
29     //*****
30     /* Ta Quang Phu - 20215450 */
31     for (int i = 0; i < n - 1; i++) {    //bubble sort
32         for (int j = i; j < n; j++) {
33             if (a[i] > a[j]){
34                 tmp = a[i];
35                 a[i] = a[j];
36                 a[j] = tmp;
37             }
38         }
39     }
40
41     //*****
42
43     printf("The sorted array is: \n");
44     for(int i = 0; i < n; i++)
45         printf("%d ", *(a + i));
46     printf("\n");
47
48     delete [] a;
49     return 0;
50 }

```

	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! ✓

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

```
int *a;
int n, tmp;

int main(){
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // #Allocate memory

    /**/
    /* Ta Quang Pho - 20215450 */
    a = (int*) malloc (n * sizeof(int));    // cấp phát bộ nhớ kích thước n phần tử kiểu
    nguyên                                // và gán địa chỉ bộ nhớ vừa cấp phát cho con trỏ a
    /**/

    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

    /**/
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < n - 1; i++) {        // bubble sort
        for (int j = i; j < n; j++) {
            if (a[i] > a[j]){
                tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
            }
        }
    }
}
```

Tạ Quang Phô - 20215450

//*****//

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

delete [] a;
return 0;
}
```

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.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void allocate_mem(int ***mt, int m, int n){
5      /*Allocate memory for the matrix
6
7      //*****//
8      /* Ta Quang Pho - 20215450 */
9      *mt = (int**) malloc (m * sizeof(int*));
10     /*cấp phát bộ nhớ m phần tử kiểu con trỏ trỏ tới kiểu nguyên
11     (cấp phát phần tử đầu tiên của các hàng)
12     gán địa chỉ bộ nhớ vừa cấp phát cho giá trị của con trỏ mt;*/
13
14     for (int i = 0; i < m; i++) {
15         (*mt)[i] = (int*) malloc (n * sizeof(int));
16         /*cấp phát bộ nhớ m phần tử kiểu nguyên
17         (cấp phát các hàng)
18         gán địa chỉ bộ nhớ vừa cấp phát cho phần tử thứ i của mảng *mt;*/
19     }
20     //*****//
21 }
22
23
24 void input(int **mt, int m, int n){
25     /*Input elements of the matrix
26
27     //*****//
28     /* Ta Quang Pho - 20215450 */
29     for (int i = 0; i < m; i++) {
30         for (int j = 0; j < n; j++) {
31             scanf("%d", &mt[i][j]);
32             /*vòng lặp để nhập lần lượt từng phần tử trong mảng
33         }
34     }
35     //*****//
36 }
37
38 void output(int **mt, int m, int n){
39     /*Print all elements of the matrix
40
41     //*****//
42     /* Ta Quang Pho - 20215450 */
43     for (int i = 0; i < m; i++) {
44         for (int j = 0; j < n; j++) {
45             printf("mt[%d][%d] = ", i, j);
46             /*in ra format của yêu cầu
47         }
48     }

```


Tạ Quang Phổ - 20215450

```

49  for (int i = 0; i < m; i++) {
50      for (int j = 0; j < n; j++) {
51          printf("%d ", mt[i][j]);
52          //vòng lặp in ra từng phần tử trong mảng
53      }
54      printf("\n");
55      //xuống dòng khi kết thúc duyệt mỗi hàng
56  }
57  return;
58  //*****//
59
60 }
61
62 int process(int **mt, int m, int n){
63     int tong = 0;
64     /* Calculate the sum of all even elements in the matrix
65
66     //*****//
67     /* Ta Quang Pho - 20215450 */
68     for (int i = 0; i < m; i++) {
69         for (int j = 0; j < n; j++) {
70             if (mt[i][j] % 2 == 0) tong += mt[i][j];
71             //kiểm tra điều kiện số chẵn, nếu thỏa mãn, tăng tổng thêm giá trị
72             //của phần tử [i][j]
73         }
74     }
75     //*****//
76
77     return tong;
78 }
79
80 void free_mem(int **mt, int m, int n){
81     /* Free memory
82
83     //*****//
84     /* Ta Quang Pho - 20215450 */
85     for (int i = 0; i < m; i++) {
86         free(mt[i]);
87         //xóa các hàng
88     }
89     free(mt);
90     //xóa mảng chứa các phần tử đầu tiên của hàng
91     return;
92     //*****//
93 }
94
95 int main(){
96     int m, n, **mt;
97     printf("Enter m, n = ");
98     scanf("%d%d", &m, &n);
99     allocate_mem(&mt, m, n);
100    input(mt, m, n);
101    output(mt, m, n);
102    printf("The sum of all even elements is %d", process(mt, m, n));
103    free_mem(mt, m, n);
104    return 0;
105 }
106

```

	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] = 1 2 4 5 The sum of all even elements is 6	✓
Passed all tests! ✓				

IT3040 – 2022.2 – Mã lớp TH: 727640

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

void allocate_mem(int ***mt, int m, int n){
    // #Allocate memory for the matrix

    /**/
    /* Ta Quang Pho - 20215450 */
    *mt = (int**) malloc (m * sizeof(int*));
    /*cấp phát bộ nhớ m phần tử kiểu con trỏ tới kiểu nguyên
    (cấp phát phần tử đầu tiên của các hàng)
    gán địa chỉ bộ nhớ vừa cấp phát cho giá trị của con trỏ mt;*/

    for (int i = 0; i < m; i++) {
        (*mt)[i] = (int*) malloc (n * sizeof(int));
        /*cấp phát bộ nhớ m phần tử kiểu nguyên
        (cấp phát các hàng)
        gán địa chỉ bộ nhớ vừa cấp phát cho phần tử thứ i của mảng *mt;*/
    }
    /**/
}

void input(int **mt, int m, int n){
    // #Input elements of the matrix

    /**/
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &mt[i][j]);
            // vòng lặp để nhập lần lượt từng phần tử trong mảng
        }
    }
    /**/
}

void output(int **mt, int m, int n){
```

Tạ Quang Phô - 20215450

```
//# Print all elements of the matrix
```

```
/**
 * Ta Quang Pho - 20215450 */
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("mt[%d][%d] = ", i, j);
        //in ra format của yêu cầu
    }
}

for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        printf("%d ", mt[i][j]);
        //vòng lặp in ra từng phần tử trong mảng
    }
    printf("\n");
    //xuống dòng khi kết thúc duyệt mỗi hàng
}

return;
/**
```

```
}
```

```
int process(int **mt, int m, int n){
    int tong = 0;
    //# Calculate the sum of all even elements in the matrix
```

```
/**
 * Ta Quang Pho - 20215450 */
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        if (mt[i][j] % 2 == 0) tong += mt[i][j];
        //kiểm tra điều kiện số chẵn, nếu thỏa mãn, tăng tổng thêm giá trị
        //của phần tử [i][j]
    }
}

/**
```

IT3040 – 2022.2 – Mã lớp TH: 727640

```
    return tong;
}

void free_mem(int **mt, int m, int n){
    // # Free memory

    /**/
    /* Ta Quang Pho - 20215450 */
    for (int i = 0; i < m; i++) {
        free(mt[i]);
        // xóa các hàng
    }
    free(mt);
    // xóa mảng chứa các phần tử đầu tiên của hàng
    return;
    /**/
}

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;
}
```

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

```

1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int n;
7
8  void printC (int* arr, int l, int r) {
9      //in ra các phần tử từ l(left) đến r(right) thuộc mảng arr
10     for (int i = l; i <= r; i++) {
11         cout << arr[i];
12     }
13     cout << endl;
14 }
15
16 int main() {
17     cout << "Ta Quang Pho - 20215450\n";
18     ifstream fin("input1.9.txt"); // File "input.txt" là file chứa bộ dữ liệu cho test case
19     if (!fin.is_open()) {
20         printf("Không mở được tệp\n"); // In thông báo nếu không mở được tệp
21         return 1;
22     }
23
24     int testCase;
25     while (fin >> testCase) {
26         cout << "Test case = " << testCase << '\n';
27         fin >> n;
28         cout << "\n" << "n = " << n;
29         int arr[n];
30         cout << "\n\n" << "Tap cha:" << '\n';
31         for (int i = 0; i < n; i++) {
32             fin >> arr[i];
33             cout << arr[i] << " ";
34         }
35         cout << "\n\n" << "Danh sach tap con:" << '\n';
36         for (int i = 0; i < n; i++) {
37             for (int j = i; j < n; j++) {
38                 printC(arr, i, j);
39                 //cho con chạy lần lượt để in các tập con theo thứ tự yêu cầu
40             }
41         }
42
43         cout << "\n\n";
44     }
45     cout << "Ta Quang Pho - 20215450";
46     return 0;
47 }
48

```

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int n;
```

```
void printC (int* arr, int l, int r) {
    //in ra các phần tử từ l(left) đến r(right) thuộc mảng arr

```

Tạ Quang Phổ - 20215450

```

for (int i = l; i <= r; i++) {
    cout << arr[i];
}
cout << endl;
}

int main() {
    cout << "Ta Quang Pho -
20215450\n_____ \n\
n";
    ifstream fin("input1.9.txt"); // File "input.txt" là file chứa bộ dữ liệu cho test case
    if (!fin.is_open()) {
        printf("Khong mo duoc tep\n"); // In thông báo nếu không mở được tệp
        return 1;
    }

    int testCase;
    while (fin >> testCase) {
        cout << "Test case = " << testCase << '\n';
        fin >> n;
        cout << "\n" << "n = " << n;
        int arr[n];
        cout << "\n\n" << "Tap cha:" << '\n';
        for (int i = 0; i < n; i++) {
            fin >> arr[i];
            cout << arr[i] << " ";
        }
        cout << "\n\n" << "Danh sach tap con:" << '\n';
        for (int i = 0; i < n; i++) {
            for (int j = i; j < n; j++) {
                printC(arr, i, j);
                //cho con chạy lần lượt để in các tập con theo thứ tự yêu cầu
            }
        }

        cout <<
"_____ \n\n";
    }
}

```

IT3040 – 2022.2 – Mã lớp TH: 727640

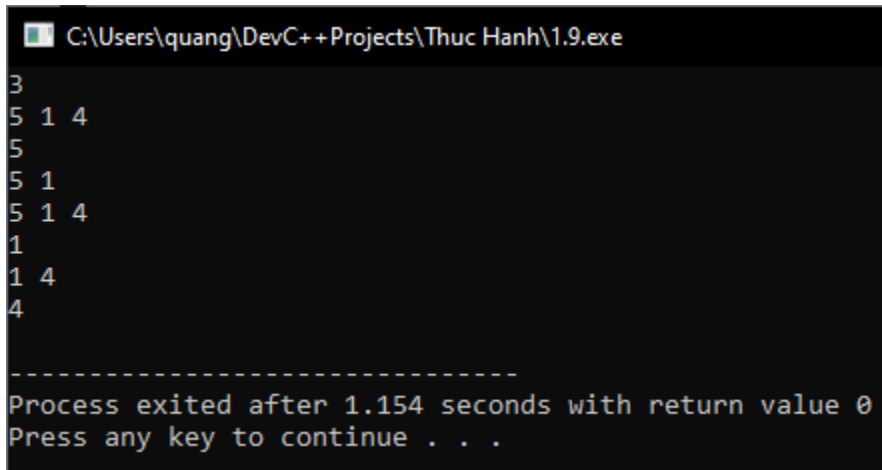
Tạ Quang Phô - 20215450

```
cout << "Ta Quang Pho - 20215450";  
return 0;  
}
```

Case 1:



Case 2:



Case 3:

Tạ Quang Phổ - 20215450

```
C:\Users\quang\DevC++\Projects\Thuc Hanh\1.9.exe
10
1 4 -2 4 65 8 -9 12 5 67
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
-2 4 65 8 -9 12 5
-2 4 65 8 -9 12 5 67
4
```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phô - 20215450

```
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
-----
Process exited after 1.34 seconds with return value 0
```

IT3040 – 2022.2 – Mã lớp TH: 727640

Case 4:

```

20
92 94 85 15 57 12 18 19 18 45 49 55 28 36 16 100 90 70 39 39
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
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

```

Tạ Quang Phổ - 20215450

```

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
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

```

Tạ Quang Phổ - 20215450

```

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
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

```

Tạ Quang Phô - 20215450

```
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
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

-----
Process exited after 16.46 seconds with return value 0
Press any key to continue . . .
```

IT3040 – 2022.2 – Mã lớp TH: 727640

Tạ Quang Phổ - 20215450

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.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int n;
5
6  // Hàm cấp phát bộ nhớ cho ma trận
7  void allocate_mem(int ***mt){
8      *mt = (int**) malloc (n * sizeof(int*)); // Cấp phát một mảng con trỏ (các phần tử đầu tiên của cột 0)
9      for (int i = 0; i < n; i++) {
10         (*mt)[i] = (int*) malloc (n * sizeof(int)); // Cấp phát bộ nhớ cho từng hàng của ma trận
11     }
12 }
13
14 // Hàm nhập ma trận từ tệp và hiển thị ma trận
15 void input(int **mt, FILE* file){
16     for (int i = 0; i < n; i++) {
17         for (int j = 0; j < n; j++) {
18             fscanf(file, "%d", (*(mt + i) + j)); // Đọc một số từ tệp và lưu vào ma trận
19             printf("%5d ", mt[i][j]); // Hiển thị giá trị đã đọc ra màn hình
20         }
21         printf("\n"); // Xuống dòng khi hoàn thành một hàng
22     }
23 }
24
25 // Hàm tính tổng hai ma trận và hiển thị kết quả
26 void addMT (int** mt1, int** mt2) {
27     for (int i = 0; i < n; i++) {
28         for (int j = 0; j < n; j++) {
29             printf("%5d ", (mt1[i][j] + mt2[i][j])); // In tổng hai phần tử [i][j] của hai ma trận
30         }
31         printf("\n"); // Xuống dòng khi hoàn thành một hàng
32     }
33 }
34
35 // Hàm nhân hai ma trận và hiển thị kết quả
36 void mulMT (int** mt1, int** mt2) {
37     int tmp;
38     for (int i = 0; i < n; i++) {

```

Tạ Quang Phổ - 20215450

```

39     for (int j = 0; j < n; j++) {
40         tmp = 0;
41         for (int k = 0; k < n; k++) {
42             tmp += mt1[i][k] * mt2[k][j]; // Tính tổng các phép nhân để tính giá trị tại [i][j]
43         }
44         printf("%5d ", tmp); // In giá trị tích [i][j] của ma trận
45     }
46     printf("\n"); // Xuống dòng khi hoàn thành một hàng
47 }
48 }
49
50 int main() {
51     printf("Ta Quang Pho - 20215450\n\n");
52     FILE *file = fopen("input1.10.txt", "r"); // File "input.txt" là file chứa bộ dữ liệu cho test case
53     if (file == NULL) {
54         printf("Khong mo duoc tep\n"); // In thông báo nếu không mở được tệp
55         return 1;
56     }
57
58     int testCase;
59     while (fscanf(file, "%d", &testCase) != EOF) {
60         printf("Test Case = %d\n", testCase);
61         int** mt1;
62         int** mt2;
63         fscanf(file, "%d", &n);
64         printf("\nn = %d\n", n); // In giá trị của n
65         allocate_mem(&mt1);
66         allocate_mem(&mt2);
67         printf("\nMa tran thu nhât:\n");
68         input(mt1, file);
69         printf("\nMa tran thu hai:\n");
70         input(mt2, file);
71         printf("\nTong hai ma tran:\n");
72         addMT(mt1, mt2);
73         printf("\nTich hai ma tran:\n");
74         mulMT(mt1, mt2);
75         printf("_____\n\n");
76     }
77
78     fclose(file);
79
80     printf("Ta Quang Pho - 20215450");
81     return 0;
82 }
83

```

#include <stdio.h>

#include <stdlib.h>

int n;

// Hàm cấp phát bộ nhớ cho ma trận

void allocate_mem(int ***mt){

 *mt = (int**) malloc (n * sizeof(int*)); // Cấp phát một mảng con trỏ (các phần tử đầu tiên của cột 0)

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

 (*mt)[i] = (int*) malloc (n * sizeof(int)); // Cấp phát bộ nhớ cho từng hàng của ma trận

 }

}

Tạ Quang Phô - 20215450

// Hàm nhập ma trận từ tệp và hiển thị ma trận

```
void input(int **mt, FILE* file){
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            fscanf(file, "%d", (*(mt + i) + j)); // Đọc một số từ tệp và lưu vào ma trận
            printf("%5d ", mt[i][j]); // Hiển thị giá trị đã đọc ra màn hình
        }
        printf("\n"); // Xuống dòng khi hoàn thành một hàng
    }
}
```

// Hàm tính tổng hai ma trận và hiển thị kết quả

```
void addMT (int** mt1, int** mt2) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%5d ", (mt1[i][j] + mt2[i][j])); // In tổng hai phần tử [i][j] của hai ma trận
        }
        printf("\n"); // Xuống dòng khi hoàn thành một hàng
    }
}
```

// Hàm nhân hai ma trận và hiển thị kết quả

```
void mulMT (int** mt1, int** mt2) {
    int tmp;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            tmp = 0;
            for (int k = 0; k < n; k++) {
                tmp += mt1[i][k] * mt2[k][j]; // Tính tổng các phép nhân để tính giá trị tại [i][j]
            }
            printf("%5d ", tmp); // In giá trị tích [i][j] của ma trận
        }
        printf("\n"); // Xuống dòng khi hoàn thành một hàng
    }
}
```

```
int main() {
```


Tạ Quang Phổ - 20215450

```
printf("Ta Quang Pho -
20215450\n_____\\n\\
n");
FILE *file = fopen("input1.10.txt", "r"); // File "input.txt" là file chứa bộ dữ liệu cho
test case
if (file == NULL) {
    printf("Khong mo duoc tep\\n"); // In thông báo nếu không mở được tệp
    return 1;
}

int testCase;
while (fscanf(file, "%d", &testCase) != EOF) {
    printf("Test Case = %d\\n", testCase);
    int** mt1;
    int** mt2;
    fscanf(file, "%d", &n);
    printf("\\nn = %d\\n", n); // In giá trị của n
    allocate_mem(&mt1);
    allocate_mem(&mt2);
    printf("\\nMa tran thu nhât:\\n");
    input(mt1, file);
    printf("\\nMa tran thu hai:\\n");
    input(mt2, file);
    printf("\\nTong hai ma tran:\\n");
    addMT(mt1, mt2);
    printf("\\nTich hai ma tran:\\n");
    mulMT(mt1, mt2);

    printf("_____\\n\\n");
}

fclose(file);

printf("Ta Quang Pho - 20215450");
return 0;
}
```

IT3040 – 2022.2 – Mã lớp TH: 727640

```

42 |         for (i = 0; i < size; i++) {
43 |             //Lap cho cac cot
44 |             for (j = 0; j < size; j++) {
45 |                 printf("%d ", matrix1[i][j] + matrix2[i][j]);
46 |             }
47 |             printf("\n");
48 |         }
49 |     }
50 | }
51 |
52 | /*
53 |     Ham nhan 2 ma tran vuong (chi in ra ket qua, khong tra ve ma tran da nhan)
54 | */
55 | void multi2Matrix (int** matrix1, int** matrix2, int size) {
56 |     //Kiem tra cac ma tran co rong hay khong va kich thuoc ma tran phai duong
57 |     if(((matrix1 != NULL) || (matrix2 != NULL)) && (size > 0)) {
58 |         int i, j, k;
59 |         int tmp;
60 |         //Lap cac hang
61 |         for (i = 0; i < size; i++) {
62 |             //Lap cho cot
63 |             for (j = 0; j < size; j++) {
64 |                 tmp = 0;
65 |                 //Lap cho phep nhan
66 |                 for (k = 0; k < size; k++){
67 |                     tmp += matrix1[i][k] * matrix2[k][j];
68 |                 }
69 |                 printf("%d ", tmp);
70 |             }
71 |             printf("\n");
72 |         }
73 |     }
74 | }
75 |
76 | /*
77 |     Ham thu hoi bo nho cap phat cho ma tran vuong
78 | */
79 | void freeMatrix(int** matrix, int size) {
80 |     //Kiem tra con tro ma tran co rong khong va size duong
81 |     if (matrix != NULL && size > 0) {
82 |         //Xoa cac hang

```

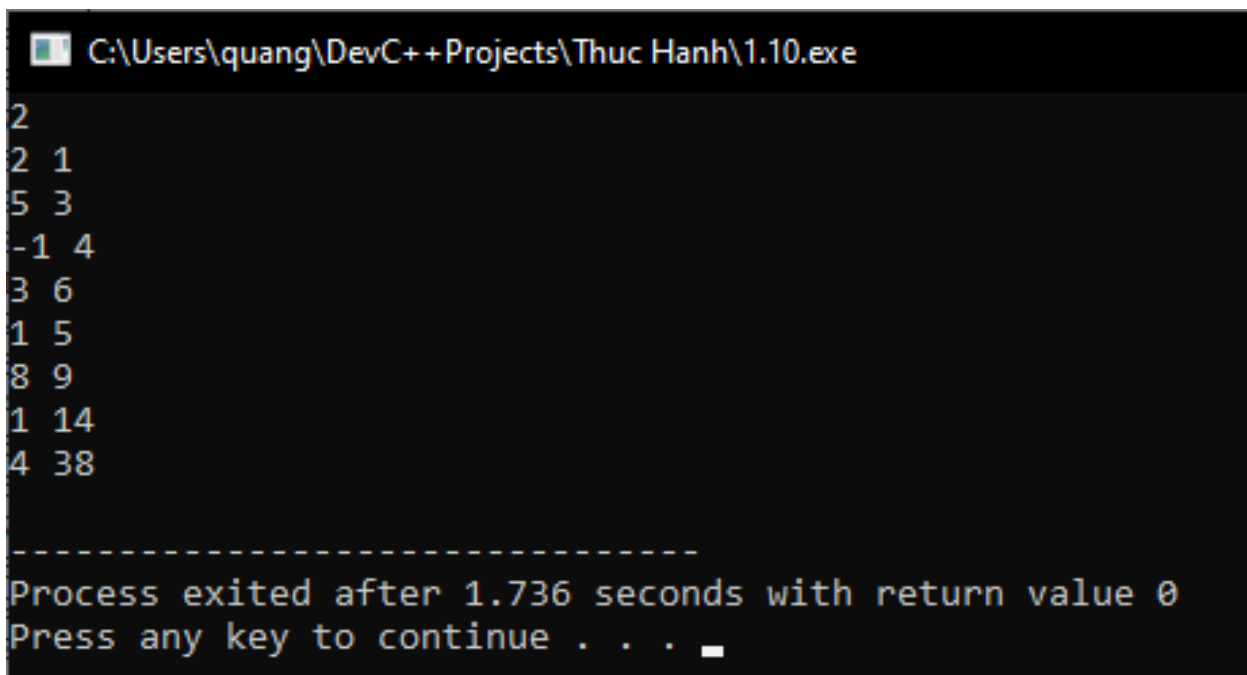
Tạ Quang Phổ - 20215450

```

83  }
84  }
85  }
86  //Xoa cot dau tien
87  free(matrix);
88  }
89  }
90
91  int main() {
92      int n;
93      scanf("%d", &n);
94      int** matrix1;
95      int** matrix2;
96      allocateMatrix (&matrix1, n);
97      enterMatrix(matrix1, n);
98      allocateMatrix (&matrix2, n);
99      enterMatrix(matrix2, n);
100     sum2Matrix (matrix1, matrix2, n);
101     multi2Matrix (matrix1, matrix2, n);
102     freeMatrix(matrix1, n);
103     freeMatrix(matrix2, n);
104     return 0;
105 }

```

Case 1:



C:\Users\quang\DevC++\Projects\Thuc Hanh\1.10.exe

```

2
2 1
5 3
-1 4
3 6
1 5
8 9
1 14
4 38
-----
Process exited after 1.736 seconds with return value 0
Press any key to continue . . .

```

IT3040 – 2022.2 – Mã lớp TH: 727640

Case 2:

```
C:\Users\quang\DevC++\Projects\Thuc Hanh\1.10.exe
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

-----
Process exited after 38.47 seconds with return value 0
Press any key to continue . . .
```

Case 3:

```

C:\Users\quang\DevC++\Projects\Thuc Hanh\1.10.exe
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
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

```