

# BÀI TẬP LỚN MÔN HỌC MÔN KĨ THUẬT LẬP TRÌNH

Đề bài: Các bài có đuôi số 5

Họ và tên: Võ Thương Trường Nhơn

Lớp: 20DTHD4

Ngày sinh: 31/10/2002

MSSV:2080600542

Email: vothuongtruongnhon2002@gmail.com

## Chủ đề 1. Mảng 1 chiều

### Bài 5.

Cho mảng một chiều gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $0 \leq n \leq 10^6$ ) và số nguyên  $m$ .

a. Cho biết mảng đã cho có bao nhiêu cặp số có tổng bằng  $m$ . Lưu ý  $(x,y)$  và  $(y,x)$  được đếm là một cặp.

b. Tính tổng bình phuong các phần tử của các cặp số có tổng bằng  $m$ .

c. Có bao nhiêu số trong mảng đã cho là bội số của  $m$ .

```

1 //Võ Thương Trường Nhơn
2 //2080600542
3 //2080600542
4 //Chu de 1 boi
5 #include<stdio.h>
6 #include<conio.h>
7 #include<stdlib.h>
8 #include<math.h>
9 #define MAX 100
10
11 void nhap (int n, int a[]);
12 void xuat(int n, int a[]);
13 int dem_tong(int n, int a[], int m);
14 int tong_binh_phuong(int n, int a[], int m);
15 int dem_boi_so(int n, int a[], int m);
16
17 int main(){
18     int n,a[MAX],m;
19     //nhap
20     printf("\nNhập số phần tử: ");
21     scanf("%d", &n);
22     nhap(n,a);
23     //xuat
24     printf("\nMảng đã nhập: ");
25     xuat(n,a);
26     printf("\n\nNhập số nguyên m: ");
27     scanf("%d",&m);
28     //a
29     int sum_A = dem_tong(n,a,m);
30     printf("\nSo cap so nguyên có tổng bằng m = %d là: %d",m,sum_A);
31     //b
32     int sum_b = tong_binh_phuong(n,a,m);
33     printf("\nTổng bình phuong của các phần tử có các cặp số có tổng bằng m = %d là: %d",m,sum_b);
34     //c
35     int sum_c = dem_boi_so(n,a,m);
36     printf("\nSo bội số của m = %d: %d",m,sum_c);
37     printf("\n\nTap something to stop!");
38     getch();
39 }
40 //nhap
41 void nhap (int n, int a[]){
42     for(int i = 0; i < n; i++){
43         printf("\nNhập a[%d]: ", i);
44         scanf("%d", &a[i]);
45     }
46 }
47 //xuat
48 void xuat(int n, int a[]){
49     for(int i = 0; i < n; i++){
50         printf("\na[%d] = %d", i, a[i]);
51     }
52 }
53 //a
54 int dem_tong(int n, int a[], int m){
55     int dem=0;
56     for(int i = 0; i < n-1; i++){
57         for(int j=i+1; j<n;j++){
58             if(m==(a[i]+a[j])){
59                 dem++;
60             }
61         }
62     }
63     return dem;
64 }
65 //b
66 int tong_binh_phuong(int n, int a[], int m){
67 }

```

```

56 int dem_dau(){
57     for(int i = 0; i < n-1; i++){
58         for(int j=i+1; j<n;j++){
59             if(m==(a[i]+a[j])){
60                 dem++;
61             }
62         }
63     }
64     return dem;
65 }
66 //b
67 int tong_binh_phuong(int n, int a[], int m){
68     int dem=0,k=0,b[101],sum;
69     for(int i = 0; i < n-1; i++){
70         for(int j=i+1; j<n;j++){
71             if(m==(a[i]*a[j])){
72                 sum+=a[i]*a[i]+a[j]*a[j];
73             }
74         }
75     }
76     return sum;
77 }
78 //c
79 int dem_boi_so(int n, int a[]){
80     int dem=0;
81     for(int i = 0; i < n; i++){
82         if(a[i] % m == 0){
83             dem++;
84         }
85     }
86     return dem;
87 }
88 //truongnhon

```

## CHỦ ĐỀ 2. MẢNG 2 CHIỀU

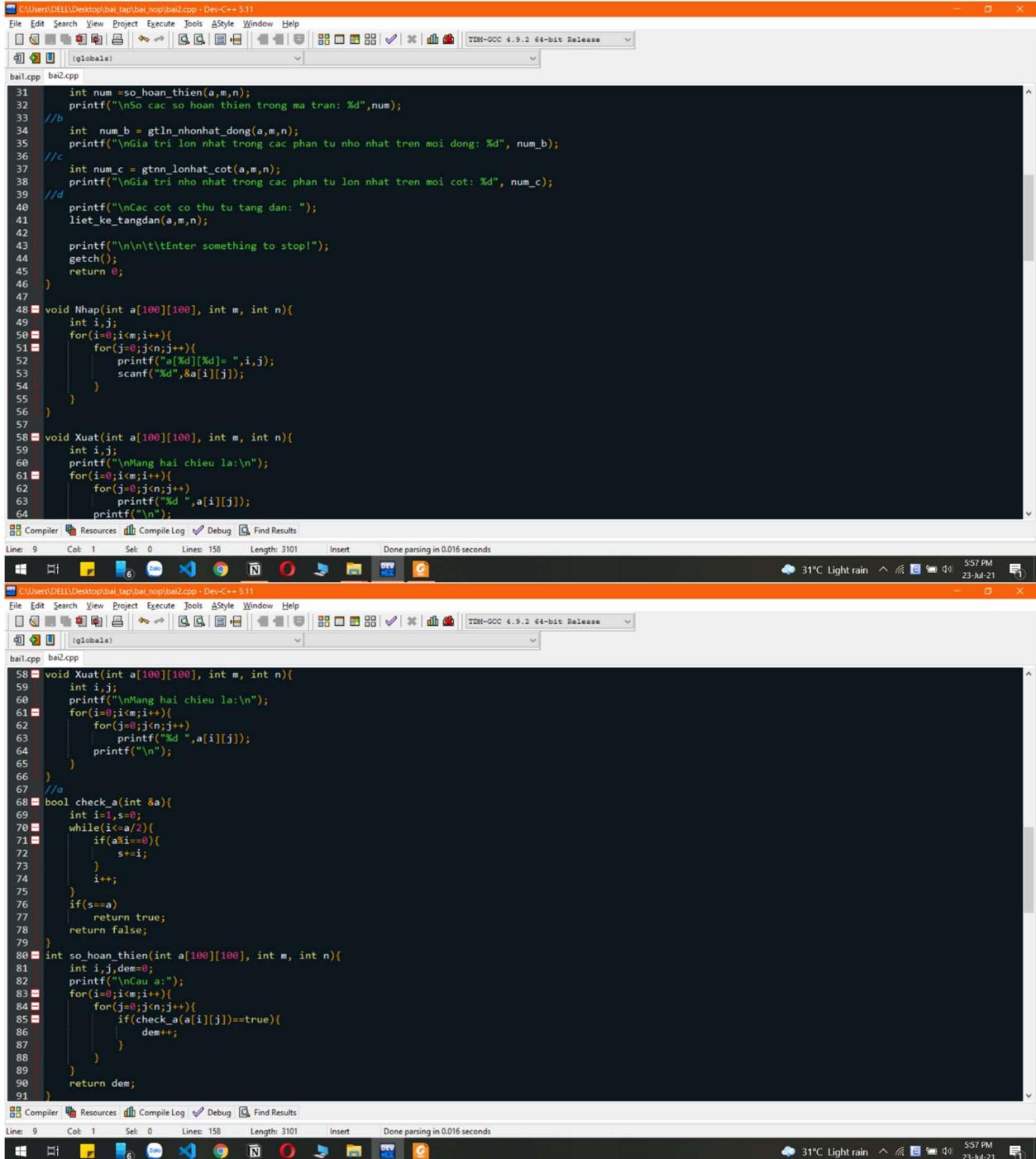
### Bài 5.

- Cho mảng A gồm  $m$  dòng  $n$  cột ( $0 < m, n \leq 200$ ), mỗi phần tử là một số nguyên.
- Có bao nhiêu số hoàn thiện trong ma trận?
  - Tìm giá trị lớn nhất trong số các phần tử nhỏ nhất của mỗi dòng ?
  - Tìm giá trị nhỏ nhất trong số các các phần tử lớn nhất trên mỗi cột ?
  - Liệt kê các cột có thứ tự tăng dần.

```

1 //Võ Thương Trường Nhơn
2 //200THD4
3 //2080600542
4 //chu de 2_bai 5
5
6 #include <stdio.h>
7 #include<conio.h>
8 #include<stdlib.h>
9
10 void Nhaph(int a[100][100], int m, int n);
11 void Xuat(int a[100][100], int m, int n);
12 bool check (int a[100][100]);
13 int so_hoan_thien(int a[100][100], int m, int n);
14 int gtm_maxhang_dong(int a[100][100], int m, int n);
15 int gtm_maxcot(int a[100][100], int m, int n);
16 bool check_cot(int a[][100], int m, int j);
17 void liet_ke_tangdan(int a[100][100], int m, int n);
18
19 int main()
20 {
21     int n,m;
22     int a[100][100];
23     //nhap
24     printf("Nhập dòng (m>0), cột(n<=200) = ");
25     scanf("%d%d",&m,&n);
26     Nhaph(a,m,n);
27     //xuat
28     Xuat(a,m,n);
29
30 //a -- 6 28 496 --
31     int num = so_hoan_thien(a,m,n);
32     printf("\nSo cac so hoan thien trong ma tran: %d",num);
33 //b
34     int num b = gtm_minhang_dong(a,m,n);

```



```

C:\Users\DELL\Desktop\bai_tap\bai_nop\bai2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
|(globals) |(globals)
bai1.cpp bai2.cpp
31     int num_so_hoan_thien(a,m,n);
32     printf("\nSo cac so hoan thien trong ma tran: %d",num);
33 //b
34     int num_b = gtn_lphnhat_dong(a,m,n);
35     printf("\nGia tri lon nhat trong cac phan tu nho nhat tren moi dong: %d", num_b);
36 //c
37     int num_c = gtnn_lonhat_cot(a,m,n);
38     printf("\nGia tri nho nhat trong cac phan tu lon nhat tren moi cot: %d", num_c);
39 //d
40     printf("\nCac cot co thu tu tang dan: ");
41     liet_ke_tangdan(a,m,n);
42
43     printf("\n\n\tEnter something to stop!");
44     getch();
45     return 0;
46 }
47
48 void Nhap(int a[100][100], int m, int n){
49     int i,j;
50     for(i=0;i<m;i++){
51         for(j=0;j<n;j++){
52             printf("%d[%d][%d] = ",i,j);
53             scanf("%d",&a[i][j]);
54         }
55     }
56 }
57
58 void Xuat(int a[100][100], int m, int n){
59     int i,j;
60     printf("\nMang hai chieu la:\n");
61     for(i=0;i<m;i++){
62         for(j=0;j<n;j++)
63             printf("%d ",a[i][j]);
64         printf("\n");
65     }
66 }
67 //a
68 bool check_a(int &a){
69     int i, s=0;
70     while(i<=a/2){
71         if(a%i==0){
72             s+=i;
73         }
74         i++;
75     }
76     if(s==a)
77         return true;
78     return false;
79 }
80
81 int so_hoan_thien(int a[100][100], int m, int n){
82     int i,j,dem=0;
83     printf("\nCau a:");
84     for(i=0;i<m;i++){
85         for(j=0;j<n;j++){
86             if(check_a(a[i][j])==true){
87                 dem++;
88             }
89         }
90     }
91     return dem;
92 }

```

Compiler Resources Compile Log Debug Find Results

Line: 9 Col: 1 Sel: 0 Lines: 158 Length: 3101 Insert Done parsing in 0.016 seconds

31°C Light rain 557 PM 23-Jul-21

```
//c
int gtnn_lonhat_cot(int a[100][100], int m, int n){
    int i,j;
    int max= a[i][0];
    int b[100],k=0;
    printf("\nCau c:");
    for(j=0;j<n;j++){
        max = a[i][0];
        for(i=0;i<m;i++){
            if(a[i][j] > max){
                max = a[i][j];
            }
        }
        b[k++] = max;
    }
    int min= b[0];
    for(i=0;i<n;i++){
        if(b[i] < min){
            min = b[i];
        }
    }
    return min;
}
//d-Liet ke cac cot co thu tu tang dan
bool check_cot(int a[][100],int m, int j){
    int i;
    for(i=0;i<m-1; i++){
        if(a[i][j]>=a[i+1][j]){
            return false;
        }
    }
    return true;
}

//b
int gtnn_nhonhat_dong(int a[100][100], int m, int n){
    int i,j;
    int min=a[i][0];
    int b[100],k=0;
    printf("\nCau b:");
    for(i=0;i<m;i++){
        min = a[i][0];
        for(j=0;j<n;j++){
            if(a[i][j] < min){
                min = a[i][j];
            }
        }
        b[k++] = min;
    }
    int max=b[0];
    for(i=0;i<n;i++){
        if(b[i] > max){
            max = b[i];
        }
    }
    return max;
}
//c
int gtnn_lonhat_cot(int a[100][100], int m, int n){
    int i,j;
    int max= a[i][0];
    int b[100],k=0;
    printf("\nCau c:");
    for(j=0;j<n;j++){
        max = a[i][0];
        for(i=0;i<m;i++){
            if(a[i][j] > max){
```

```

126     }
127 }
128 b[k++] = max;
129 }
130 int min=b[0];
131 for(i=0;i<n;i++){
132     if(b[i] < min){
133         min = b[i];
134     }
135 }
136 return min;
137 }
138 //d-liet ke cac cot co thu tu tang dan
139
140 bool check_cot(int a[][100],int m, int j){
141     int i;
142     for(i=0;i<m-1; i++){
143         if(a[i][j]>a[i+1][j]){
144             return false;
145         }
146     }
147     return true;
148 }
149 void liet_ke_tangdan(int a[100][100], int m, int n){
150     for(int i=0;i<n;i++){
151         if(check_cot(a,m,i)){
152             printf(" %d ",i);
153         }
154     }
155 }
156
157
158 //truongnhon

```

## CHỦ ĐỀ 3. CON TRỎ

### Bài 5.

Cho mảng hai chiều  $a$  gồm  $m$  dòng  $n$  cột, mỗi phần tử là một số nguyên dương.

- Đếm số lượng số hoàn chỉnh trong ma trận.
- Tính tổng các phần tử trên mỗi dòng.

```

1 //Võ Thương Trường Nhơn
2 //200TH04
3 //2080600542
4 //Chu de [ not S
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <conio.h>
9
10 void nhap(int **a, int m, int n);
11 void xuat(int **a, int m, int n);
12 bool check_cot(int *a);
13 int dem_hoan_chinh(int **a, int m, int n);
14 void tong_dong(int **a, int m, int n);
15
16 int main()
17 {
18     int **a,m,n;
19     printf("Nhập dòng(m), cột(n) = ");
20     scanf("%d%d",&m,&n);
21     a = (int **)malloc(m * sizeof(int *));//các dòng
22     for (i = 0; i < m; i++)
23     {
24         *(a+i) = (int *)malloc(n*sizeof(int *));//các cột
25     }
26     nhap(a,m,n);
27     xuat(a,m,n);
28     //a
29     int num_a = dem_hoan_chinh(a,m,n);
30     printf("\nSố lượng số hoàn chỉnh: %d", num_a);
31     //b
32     printf("\nTổng các phần tử trên mỗi dòng:");
33     tong_dong(a,m,n);
34
35     printf("\nEnter something to stop!");
36     getch();
37     return 0;
38 }
39
40 void nhap(int **a, int m, int n){
41     for (int i=0; i<m; i++){
42         for (int j=0; j<n; j++){
43             printf("a[%d][%d]=",i,j);
44             scanf("%d",&a[i][j]);
45         }
46     }
47 }

```

The image shows two windows of the Dev-C++ IDE running on a Windows operating system. Both windows have the title bar "C:\Users\DELL\Desktop\bai\_tap\bai3.cpp - Dev-C++ 5.11". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help. The toolbar contains icons for New, Open, Save, Print, Run, Stop, and others. The status bar at the bottom shows Line: 4 Cob: 11 Sel: 0 Lines: 98 Length: 1743 Insert Done parsing in 0.016 seconds. The system tray shows the date and time as 7:38 AM 20-Jul-21.

Code in the left window (lines 40-82):

```
40 void nhap(int **a, int m, int n){  
41     for (int i=0; i<m; i++){  
42         for (int j=0; j<n; j++){  
43             printf("%d[%d][%d]=", i, j);  
44             scanf("%d", &(a[i][j]));  
45         }  
46     }  
47 }  
48  
49 void xuat(int **a, int m, int n){  
50     printf("Cac phan tu cua mang:\n");  
51     for (int i=0; i<m; i++){  
52         for (int j=0; j<n; j++){  
53             printf("%d", *(a[i]+j));  
54         }  
55         printf("\n");  
56     } //a  
57 }  
58 bool check_a(int *a){  
59     int i=1, s=0;  
60     while(i<a/2){  
61         if(a[i]==0){  
62             s+=i;  
63         }  
64         i++;  
65     }  
66     if(s==a){  
67         return true;  
68     }else{  
69         return false;  
70     }  
71 }  
72  
73 int dem_hoan_chinh(int **a, int m, int n){  
74     int dem=0;  
75     for (int i=0; i<m; i++){  
76         for (int j=0; j<n; j++){  
77             if(check_a(*(a+i)+j))==true){  
78                 dem++;  
79             }  
80         }  
81     }  
82     return dem;  
83 }  
84 //b  
85 void tong_dong(int **a, int m, int n){  
86     int s=0;  
87     int *s1;  
88     int k=0;  
89     for (int i=0; i<m; i++){  
90         s=0;  
91         for (int j=0; j<n; j++){  
92             s+=a[i][j];  
93         }  
94         printf("%d ", s);  
95     }  
96 }  
97 }  
98 }
```

Code in the right window (lines 56-98):

```
56 }  
57 } //a  
58 bool check_a(int *a){  
59     int i=1, s=0;  
60     while(i<a/2){  
61         if(a[i]==0){  
62             s+=i;  
63         }  
64         i++;  
65     }  
66     if(s==a){  
67         return true;  
68     }else{  
69         return false;  
70     }  
71 }  
72  
73 int dem_hoan_chinh(int **a, int m, int n){  
74     int dem=0;  
75     for (int i=0; i<m; i++){  
76         for (int j=0; j<n; j++){  
77             if(check_a(*(a+i)+j))==true){  
78                 dem++;  
79             }  
80         }  
81     }  
82     return dem;  
83 }  
84 //b  
85 void tong_dong(int **a, int m, int n){  
86     int s=0;  
87     int *s1;  
88     int k=0;  
89     for (int i=0; i<m; i++){  
90         s=0;  
91         for (int j=0; j<n; j++){  
92             s+=a[i][j];  
93         }  
94         printf("%d ", s);  
95     }  
96 }  
97 }  
98 }
```

## CHỦ ĐỀ 4. ĐỆ QUY

### Bài 5.

Dãy số  $x_n$  định nghĩa như sau:

$$x_1 = 1;$$

$$x_2 = 2;$$

$$x_3 = 3;$$

$$x_n = 2 \times x_{n-1} \times (5 \times x_{n-2} + 3 \times x_{n-3}) \text{ với mọi } n \geq 4$$

a. Từ định nghĩa trên, hãy cho biết giá trị  $x_7$ .

b. Viết hàm tính  $x_n$  bằng cách sử dụng đệ quy.

c. Viết hàm tính  $x_n$  bằng cách không sử dụng đệ quy.

### Bài làm:

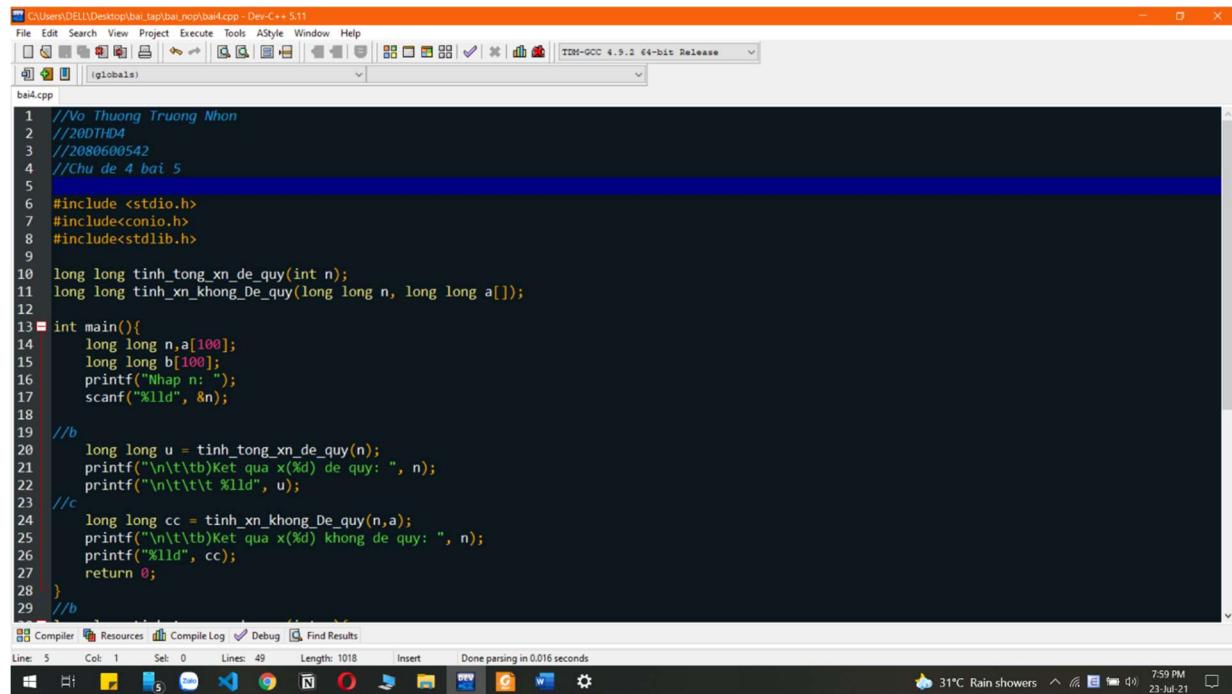
a)

$$a[4] = 78$$

$$a[5] = 3276$$

$$a[6] = 2614248$$

$$a[7] = 86866232544 (\text{Giá trị cần tính})$$



The screenshot shows the Dev-C++ IDE interface with the file "bai4.cpp" open. The code is written in C++ and defines three functions: tinh\_tong\_xn\_de\_quy, tinh\_xn\_khong\_De\_quy, and main. The main function prompts the user for an input n, calls tinh\_tong\_xn\_de\_quy to get the result, and then prints it. The code uses long long type for large numbers and includes stdio.h, conio.h, and stdlib.h headers. The IDE shows syntax highlighting and a status bar at the bottom indicating the parsing time.

```

C:\Users\DELL\Desktop\bai_tap\bai_nop\bai4.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
File Project View Tools Window Help
bai4.cpp
21     printf("\n\t\tKết quả x(%d) de quy: ", n);
22     printf("\n\t\t\t %lld", u);
23 }
24 //c
25 long long cc = tinh_xn_khong_De_quy(n,a);
26 printf("\n\t\tKết quả x(%d) khong de quy: ", n);
27 printf("%lld", cc);
28 return 0;
29 //}
30 long long tinh_tong_xn_de_quy(int n){
31 if(n<=3){
32     return n;
33 }
34 return 2 * tinh_tong_xn_de_quy(n-1) * (5 * tinh_tong_xn_de_quy(n-2) + 3 * tinh_tong_xn_de_quy(n-3));
35 }
36 //}
37 long long tinh_xn_khong_De_quy(long long n, long long a[]){
38     a[1]=1,a[2]=2,a[3]=3;
39     long long m;
40     if(n<=3){
41         return n;
42     }else{
43         for(int i=4;i<=n;i++){
44             a[i] = 2*a[i-1]*(5*a[i-2]+3*a[i-3]);
45             m=a[i];
46         }
47     }
48     return m;
49 }

```

## CHỦ ĐỀ 5. FILE

### Bài 5.

Cho mảng hai chiều  $a$  có  $m$  dòng và  $n$  cột ( $m, n \leq 200$ ); các phần tử là các số nguyên dương. Hãy viết chương trình hoàn chỉnh thực hiện các công việc sau:

a. Đếm xem mảng có bao nhiêu số nguyên tố ?

b. Cho biết giá trị lớn nhất trong số các tổng các phần tử trên mỗi dòng.

Dữ liệu vào được cho từ file văn bản table.inp; trong đó

- Dòng đầu ghi 2 số  $m,n$

- Trong  $m$  dòng tiếp theo mỗi dòng ghi  $n$  số; mỗi số cách nhau ít nhất một khoảng trắng.

Kết quả ghi vào file văn bản table.out gồm 2 dòng; mỗi dòng ghi kết quả của mỗi câu tương ứng.

Ví dụ:

TABLE.INP

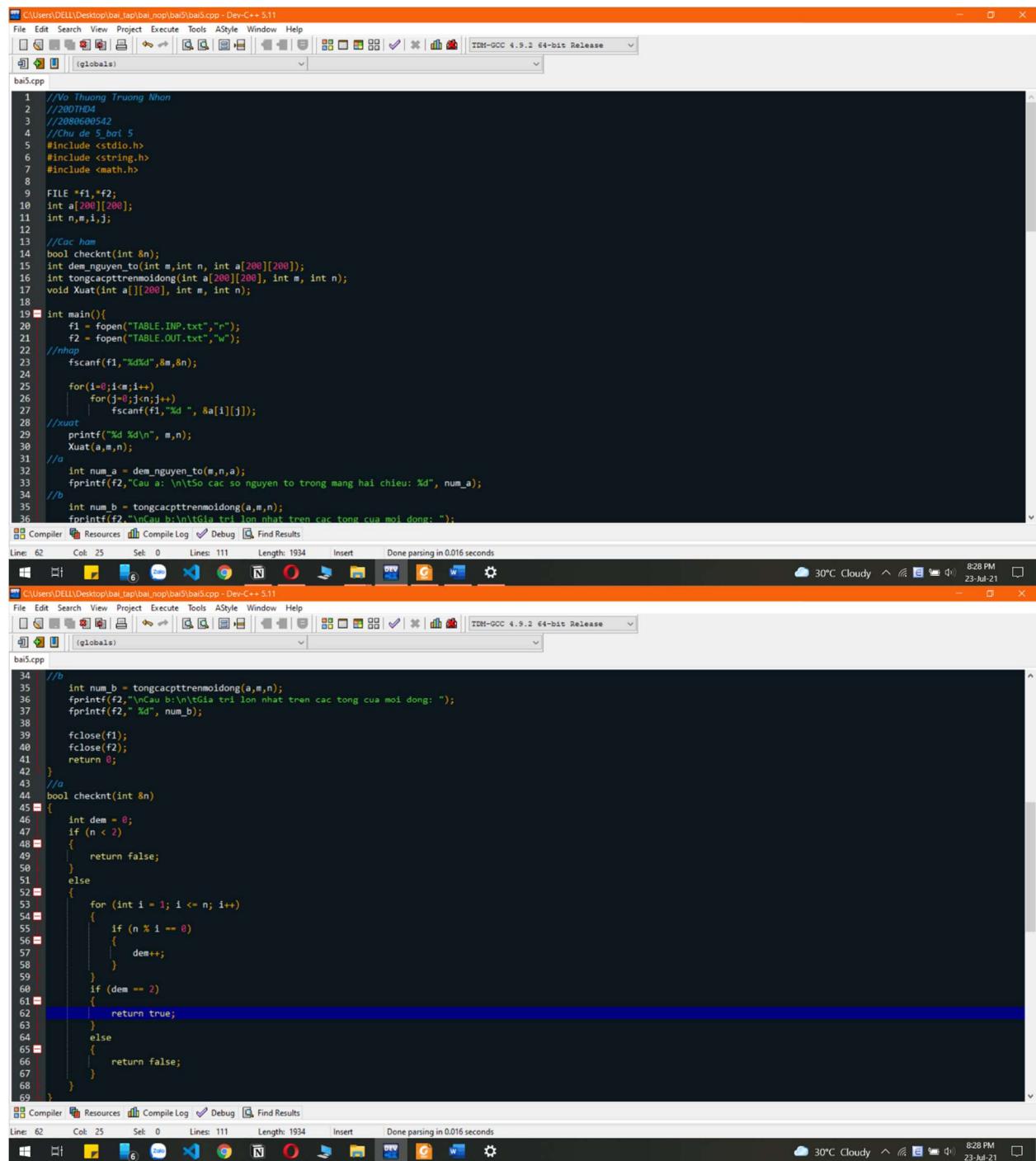
4 5

3	4	2	2	1
5	10	101	13	10
4	20	31	101	3
101	31	10	101	7

TABLE.OUT

Câu a: 13

Câu b: 250



```
1 //Võ Thương Trường Nhơn
2 //200TH04
3 //2080600542
4 //Chu de 5_bài 5
5 #include <stdio.h>
6 #include <string.h>
7 #include <math.h>
8
9 FILE *f1,*f2;
10 int a[200][200];
11 int m,n,i,j;
12
13 //Car ham
14 bool checknt(int &n);
15 int dem_nguyen_to(int m,int n, int a[200][200]);
16 int tongcacptrremoidong(int a[200][200], int m, int n);
17 void Xuat(int a[][200], int m, int n);
18
19 int main(){
20     f1 = fopen("TABLE.INP.txt","r");
21     f2 = fopen("TABLE.OUT.txt","w");
22     //nhap
23     fscanf(f1,"%d%d",&m,&n);
24
25     for(i=0;i<m;i++)
26         for(j=0;j<n;j++)
27             fscanf(f1,"%d ", &a[i][j]);
28     //xuat
29     printf("%d %d\n", m,n);
30     Xuat(a,m,n);
31     //a
32     int num_a = dem_nguyen_to(m,n,a);
33     fprintf(f2,"Cau a: \n\tSo cac so nguyen to trong mang hai chieu: %d", num_a);
34     //b
35     int num_b = tongcacptrremoidong(a,m,n);
36     fprintf(f2,"Cau b:\n\tbia tri lon nhat tren cac tong cua moi dong: %d", num_b);
37
38     fclose(f1);
39     fclose(f2);
40     return 0;
41 }
42 //a
43 bool checknt(int &n)
44 {
45     int dem = 0;
46     if (n < 2)
47     {
48         return false;
49     }
50     else
51     {
52         for (int i = 1; i <= n; i++)
53         {
54             if (n % i == 0)
55             {
56                 dem++;
57             }
58             if (dem == 2)
59             {
60                 return true;
61             }
62             else
63             {
64                 return false;
65             }
66         }
67     }
68 }
```

The code is a C++ program named bai5.cpp. It reads a 2D integer array 'a' from a file TABLE.INP.txt and writes its properties to TABLE.OUT.txt. The program includes functions to calculate the sum of elements in each row ('tongcacptrremoidong') and to check if a number is prime ('checknt'). It also includes a function 'Xuat' to print the matrix.

In the second window, the 'checknt' function has been moved to a separate header file named 'checknt.h'. The code in bai5.cpp now includes the header file and calls the function from there.

```
34 //b
35     int num_b = tongcacptrremoidong(a,m,n);
36     fprintf(f2,"Cau b:\n\tbia tri lon nhat tren cac tong cua moi dong: %d", num_b);
37
38     fclose(f1);
39     fclose(f2);
40     return 0;
41 }
42 //a
43 bool checknt(int &n)
44 {
45     int dem = 0;
46     if (n < 2)
47     {
48         return false;
49     }
50     else
51     {
52         for (int i = 1; i <= n; i++)
53         {
54             if (n % i == 0)
55             {
56                 dem++;
57             }
58             if (dem == 2)
59             {
60                 return true;
61             }
62             else
63             {
64                 return false;
65             }
66         }
67     }
68 }
```

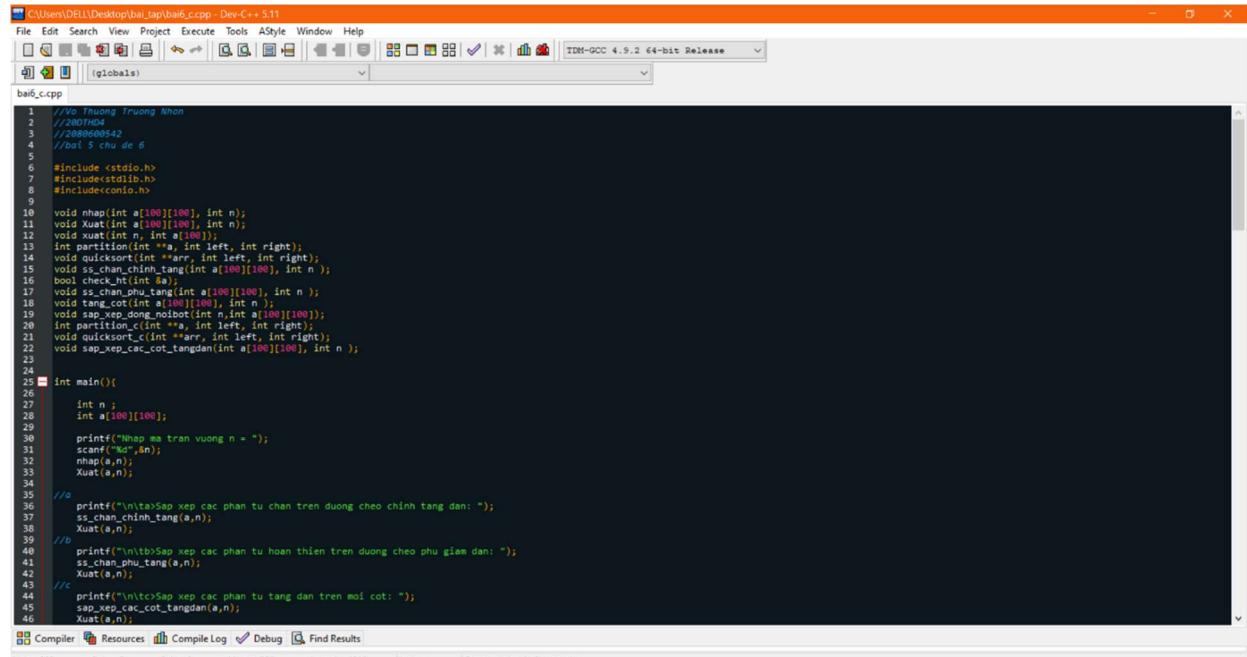
```
71 int dem_nguyen_to(int m, int n, int a[200][200]){
72     int dem=0;
73     for(i=0;i<m;i++){
74         for(j=0;j<n;j++){
75             if(checknt(a[i][j])){
76                 dem++;
77             }
78         }
79     }
80     return dem;
81 }
83 //b
84 int tongcacptrentrmoidong(int a[200][200], int m, int n){
85     int i,j,sum=0;
86     int b[100],k=0;
87     int max=b[0];
88     for(i=0;i<m;i++){
89         for(j=0;j<n;j++){
90             sum+=a[i][j];
91         }
92         b[k++]=sum;
93         sum=0;
94     }
95     for(i=0;i<k;i++){
96         if(b[i] > max){
97             max = b[i];
98         }
99     }
100    return max;
101 }
102 void Xuat(int a[][200], int m, int n){
103     int i,j;
104     printf("\nMảng hai chiều là:\n");
105     for(i=0;i<m;i++){
106         for(j=0;j<n;j++){
107             printf("%d ",a[i][j]);
108             printf("\n");
109         }
110     }
111 }
```

## CHỦ ĐỀ 6. TÌM KIẾM VÀ SẮP XẾP

### Bài 5. Quick Sort

Cho mảng  $a$  gồm  $n$  dòng  $n$  cột, mỗi phần tử là một số nguyên.

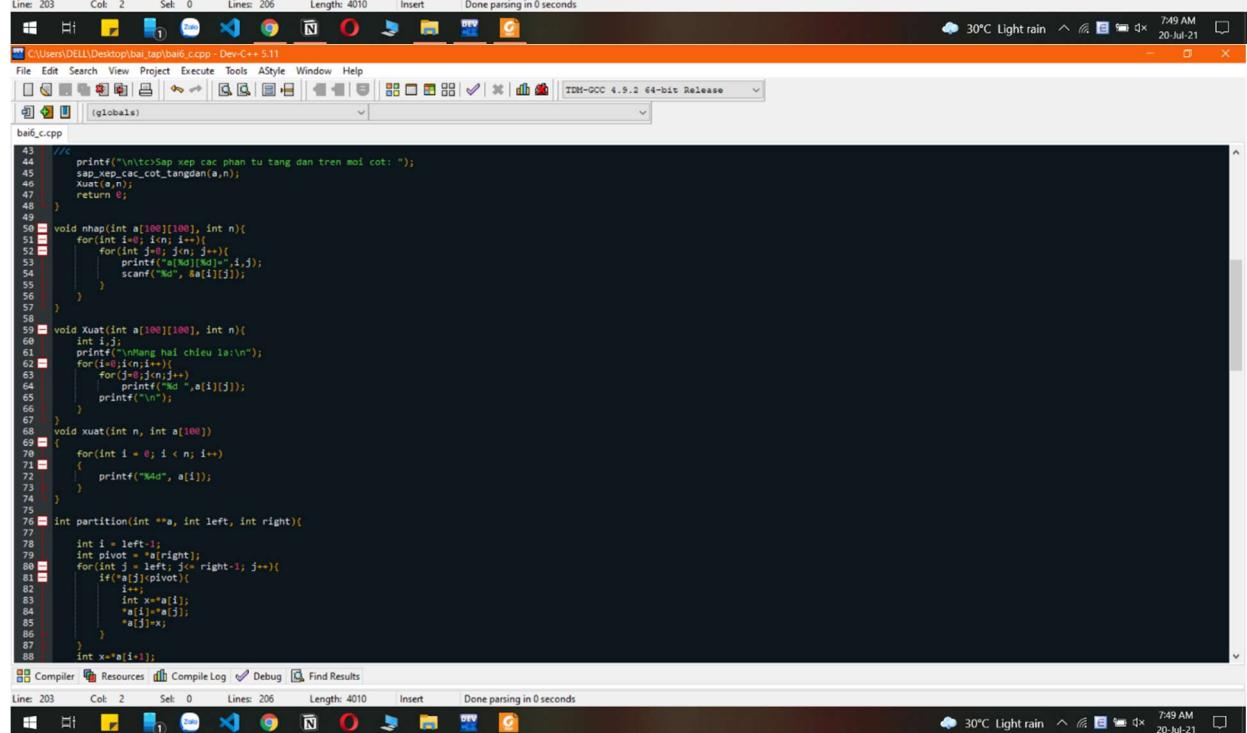
- a. Sắp xếp các phần tử chẵn trên đường chéo chính tăng dần.
- b. Sắp xếp các phần tử hoàn thiện trên đường chéo phụ giảm dần.
- c. Sắp xếp các phần tử tăng dần trên mỗi cột.



```

1 //Võ Thương Trường Nhơn
2 //2080600542
3 //2080600542
4 //bai6_5 chia de 6
5
6 #include <stdio.h>
7 #include<stdlib.h>
8 #include<conio.h>
9
10 void nhap(int a[100][100], int n);
11 void Xuat(int a[100][100], int n);
12 void Jaso(int n, int m[100]);
13 int partition(int **a, int left, int right);
14 void quicksort(int **arr, int left, int right);
15 void ss_chinh_chinh_tang(int a[100][100], int n );
16 void check_bt(int a[100][100], int n );
17 void sap_xep_duoi_noibot(int a[100][100], int n );
18 void tang_cot(int a[100][100], int n );
19 void sap_xep_dong_noibot(int n,int a[100][100]);
20 int partition_c(int **a, int left, int right);
21 void quicksort_c(int **arr, int left, int right);
22 void sap_xep_cac_cot_tangdan(int a[100][100], int n );
23
24
25 int main(){
26
27     int n ;
28     int a[100][100];
29
30     printf("Nhap ma tran vuong n = ");
31     scanf("%d",&n);
32     nhap(a,n);
33     Xuat(a,n);
34
35     //a
36     printf("\n(a) Sap xep cac phan tu chan tren duong cheo chinh tang dan: ");
37     ss_chinh_chinh_tang(a,n);
38     Xuat(a,n);
39
40     //b
41     printf("\n(b) Sap xep cac phan tu hoan thien tren duong cheo phu giam dan: ");
42     ss_chinh_phu_tang(a,n);
43     Xuat(a,n);
44
45     //c
46     printf("\n(c) Sap xep cac phan tu tang dan tren moi cot: ");
47     sap_xep_cac_cot_tangdan(a,n);
48     Xuat(a,n);
49 }

```



```

53     printf("%d",a[i][j]);
54     scanf("%d", &a[i][j]);
55 }
56 }
57 }
58
59 void nhap(int a[100][100], int n){
60     for(int i=0; i<n; i++){
61         for(int j=0; j<n; j++){
62             printf("%d",a[i][j]);
63             scanf("%d", &a[i][j]);
64         }
65     }
66 }
67
68 void Xuat(int a[100][100], int n)
69 {
70     for(int i = 0; i < n; i++)
71     {
72         printf("%d", a[i]);
73     }
74 }
75
76 int partition(int **a, int left, int right){
77
78     int l = left-1;
79     int pivot = *a[right];
80
81     for(int j = left; j< right-1; j++){
82         if(a[j]<pivot){
83             i++;
84             int x=a[i];
85             a[i]=a[j];
86             a[j]=x;
87         }
88     }
89     int x=a[i+1];
90
91 }

```

```

76 int partition(int **a, int left, int right){
77     int i = left-1;
78     int pivot = *(right);
79     for(int j = left; j <= right-1; j++){
80         if(*a[j] > pivot){
81             i++;
82             int x=a[i];
83             *a[i]=*a[j];
84             *a[j]=x;
85         }
86     }
87     int x=a[i+1];
88     *a[i+1]=*(right);
89     *(right)=x;
90     return (i+1);
91 }
92 void quicksort(int **arr, int left, int right){
93     if (left < right){
94         int pi = partition(arr, left, right);
95         quicksort(arr, left, pi - 1);
96         quicksort(arr, pi + 1, right);
97     }
98 }
99 //void ss_chinh_chinh_tang(int a[100][100], int n ){
100 int i,j,k=0;
101 int **b = new int*[n];
102 for(i=0;i<n;i++){
103     b[i] = new int[n];
104     for(j=0;j<n;j++){
105         if(a[i][j]<=a[k]){
106             b[k+j]=a[i][j];
107         }
108     }
109 }
110 quicksort(b,0,K-1);
111 }
112 //void ss_chinh_phu_tang(int a[100][100], int n ){
113 int partition_b(int **a, int left, int right){
114     int i = left-1;
115     int pivot = *(right);
116     for(int j = left; j <= right-1; j++){
117         if(*a[j] > pivot){
118             i++;
119             int x=a[i];
120             *a[i]=*a[j];
121             *a[j]=x;
122         }
123     }
124     int x=a[i+1];
125     *a[i+1]=*(right);
126     *(right)=x;
127     return (i+1);
128 }
129 void quicksort_(int **arr, int left, int right){
130     if (left < right){
131         int pi = partition_b(arr, left, right);
132         quicksort_(arr, left, pi - 1);
133         quicksort_(arr, pi + 1, right);
134     }
135 }
136 bool check_ht(int a[]){
137     int i,s=0;
138     while(i<s/2){
139         if(a[i]==0){
140             s+=1;
141         }
142         i++;
143     }
144     if(s==e)
145         return true;
146     return false;
147 }
148 void ss_chinh_phu_tang(int a[100][100], int n ){
149 int i,j,k=0;
150 int **b = new int*[n];
151 for(i=0;i<n;i++){
152

```

```

151     } return false;
152 }
153
154 void ss_chuan_phu_tang(int a[100][100], int n ){
155     int i,j,k=0;
156     int **b = new int*[n];
157     for(i=0;i<n;i++){
158         if(check_ht(a[i][i][n-1-i])){
159             b[k++]=a[i][i][n-1-i];
160         }
161     }
162     quicksort_b(b,0,k-1);
163 }
164 */
165
166 int partition_c(int **a, int left, int right){
167
168     int i = left-1;
169     int pivot = *a[right];
170     for(int j = left; j<= right-1; j++){
171         if(*a[j]<pivot){
172             i++;
173             *a[i]=*a[j];
174             *a[j]=*a[i];
175             *a[i]=x;
176         }
177     }
178     int x=a[i+1];
179     *a[i+1]=*a[right];
180     *a[right]=x;
181     return (i+1);
182 }
183 void quicksort_c(int **arr, int left, int right){
184
185     if (left < right){
186
187         int pi = partition(arr, left, right);
188         quicksort(arr, left, pi - 1);
189         quicksort(arr, pi + 1, right);
190     }
191 }
192
193 void sap_xep_cac_cot_tangdan(int a[100][100], int n ){
194     int i,j,k=0;
195     int **b = new int*[n];
196     for(j=0;j<n;j++){
197
198         if(check_ht(a[i][j][n-1-j])){
199             b[k++]=a[i][j][n-1-j];
200         }
201     }
202     quicksort_c(b,0,k-1);
203     k = 0;
204 }

```

## CHỦ ĐỀ 7. TỔNG HỢP

**Bài 5:** Viết chương trình quản lý lớp học của một trường. Các thông tin của một lớp học như sau: tên

lớp, số lượng giáo viên chủ nhiệm, số lượng học sinh giỏi, số lượng học sinh khá, số lượng

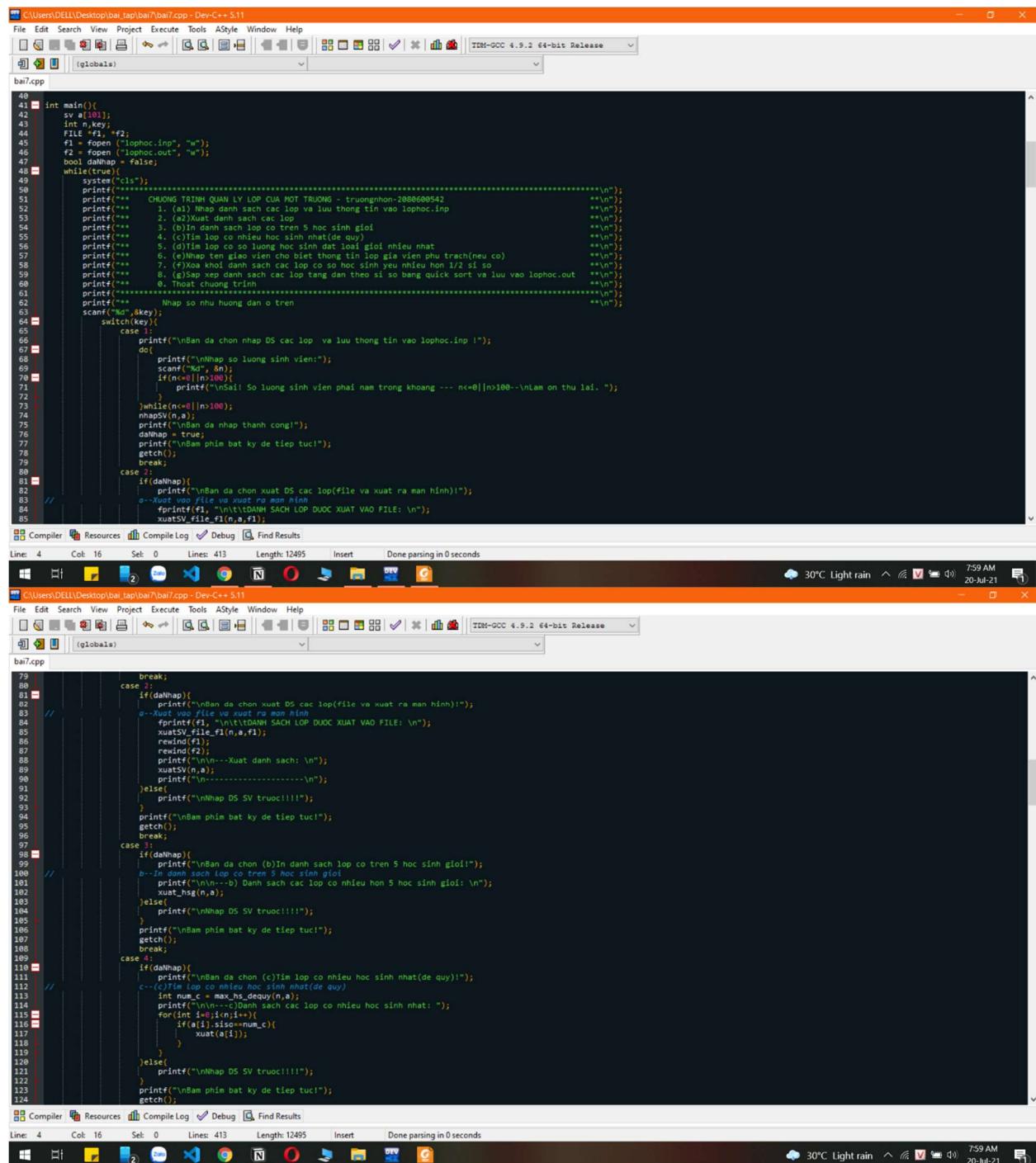
hoc

sinh trung bình, số lượng học sinh yếu.

- a. Nhập vào danh sách các lớp với thông tin yêu cầu như trên và lưu vào tập tin lophoc.inp
- b. In danh sách các lớp có trên 5 học sinh giỏi.
- c. Tim lớp có nhiều học sinh nhất bằng kỹ thuật đệ quy
- d. Tim lớp có số lượng học sinh đạt loại giỏi nhiều nhất.
- e. Nhập tên của giáo viên, hãy cho biết thông tin của lớp do giáo viên đó phụ trách (nếu có).
- f. Xóa khỏi danh sách các lớp có số lượng học sinh yếu nhiều hơn  $\frac{1}{2}$  sĩ số.
- g. Sắp xếp danh sách các lớp tăng dần theo sĩ số bằng phương pháp sắp xếp nhanh (quick sort) và lưu vào tập tin lophoc.out

The screenshot shows the Dev-C++ IDE interface with the file 'bai7.cpp' open. The code implements various functions for managing student data (sv) and class lists (lolop). It includes functions for reading from CSV files, searching for students by name or ID, finding the class with the most students, and sorting classes by student count using quicksort. The code uses structures, arrays, and standard C/C++ libraries like stdio.h and string.h.

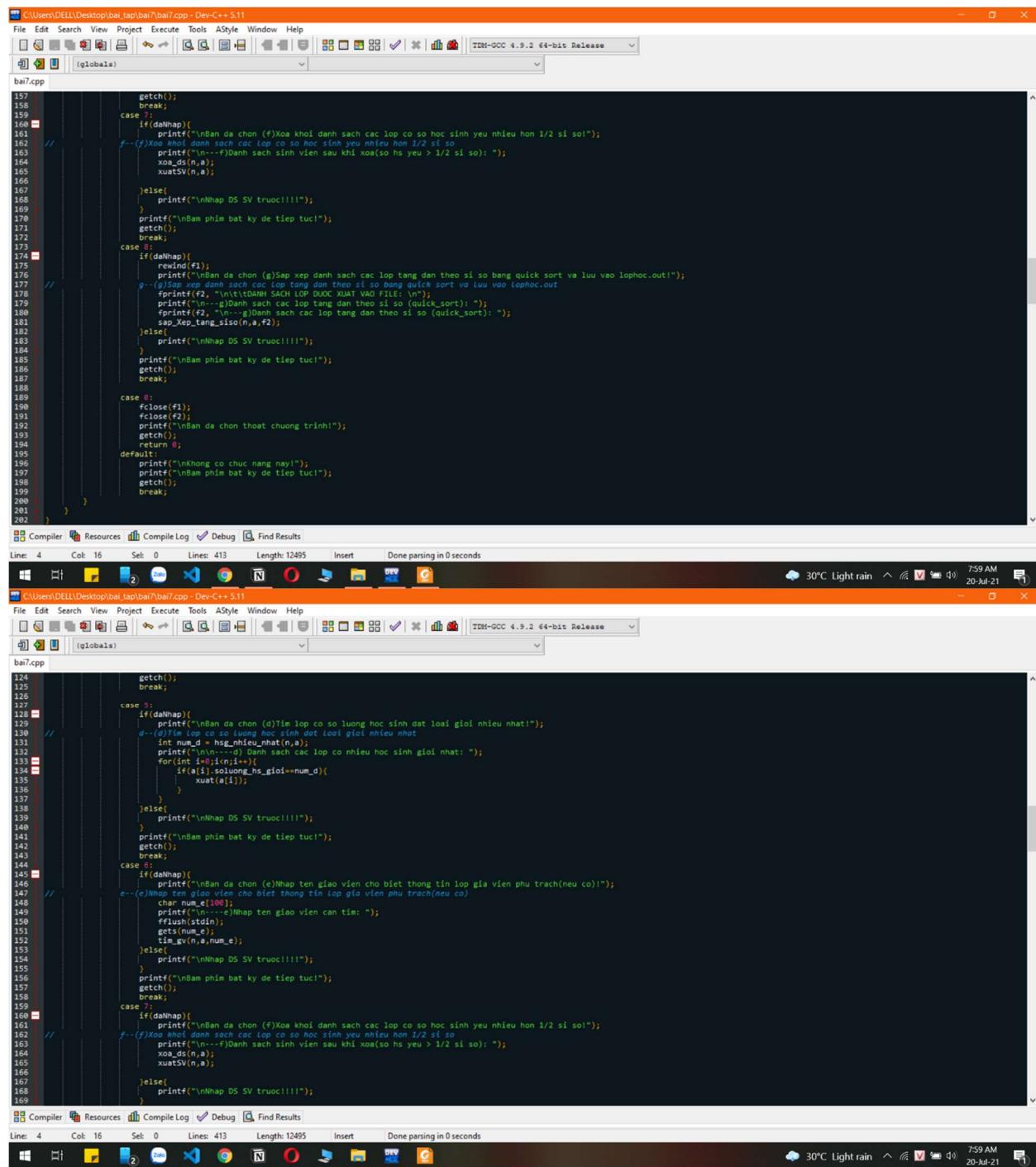
```
1 //Võ Thương Trường Nhơn
2 //2807H04
3 //2080600542
4 //CHƯƠNG TRÌNH TÌM LỚP CÓ HỌC SINH NHẤT BẰNG KỸ THUẬT ĐỆ QUY
5 #include <iostream.h>
6 #include<stdlib.h>
7 #include<string.h>
8
9
10 struct sinhvien{
11     char tenlop[10];
12     int siso;
13     char hoten_gvc[50];
14     int soluong_hs_gioi;
15     int soluong_hs_kha;
16     int soluong_hs_trungbinh;
17     int soluong_hs_yeu;
18 };
19
20 typedef struct sinhvien sv;
21
22 void nhapSv(int n, sv a[]);
23 void xuat(sv x);
24 void xuatSv(int n, sv a[]);
25 void xuatSV(int n, sv a[], FILE *f1);
26 void xuat_xv(FILE *xv, FILE *f1);
27 void xuat_file_f2(sv x, FILE *f2);
28 void xuatSV_file_f1(int n, sv a[],FILE *f1);
29 void xuatSV_file_f2(int n, sv a[],FILE *f2);
30 void xuat_hsg(int n, sv a[]);
31 void xuat_hsg_sv(int n, sv a[]);
32 void xuat_hsg_nhom(int n, sv a[]);
33 void xuat_hsg_nhom_sv(int n, sv a[]);
34 void xoa_ds(int &n, sv a[]);
35 void swap(int &a, int &b);
36 int partition(sv a[], int left, int right);
37 void quicksort(sv a[], int left, int right);
38 int tim_kiem(int n, sv a[], int b,FILE *f2);
39 void sap_xep_tang_siso(int n, sv a[], FILE *f2);
40
41 int main(){
42     sv a[100];
43     int n,Key;
44     FILE *f1, *f2;
45     f1 = fopen ("lophoc.inp", "r");
46     f2 = fopen ("lophoc.out", "w");
47 }
```



```

40
41 int main(){
42     sv a[100];
43     int n,key;
44     FILE *f1,*f2;
45     f1 = fopen ("lophoc.inp", "w");
46     f2 = fopen ("lophoc.out", "w");
47     bool daNhap = false;
48
49     while(true){
50         system("cls");
51         printf("***** CHUC NAM QUAN LY LOP CUA MOT TRUONG - truongnhon-2080600542 *****\n");
52         printf("1. (a) Nhap danh sach cac lop va luu thong tin vao lophoc.inp\n");
53         printf("2. (b) Xuat danh sach cac lop\n");
54         printf("3. (b)In danh sach lop co tren 5 hoc sinh gioi\n");
55         printf("4. (c)Tim lop co nhieu hoc sinh nhat(de quy)\n");
56         printf("5. (d)Tim lop co so luong hoc sinh dat lop gioi nhieu nhat\n");
57         printf("6. (e)Xuat ten cua cac lop co so luong hoc sinh dat lop gioi nhieu nhat\n");
58         printf("7. (f)Xoa khong danh sach cac lop co so hoc sinh yeu nhieu hon 1/2 sỉ so\n");
59         printf("8. (g)Xep danh sach cac lop tang dan theo so bang quick sort va luu vao lophoc.out\n");
60         printf("9. Thoat chuong trinh\n");
61         printf("Nhập số nhu huống dàn trên\n");
62         scanf("%d",&key);
63     }
64     switch(key){
65     case 1:
66         printf("\nBạn đã chọn nhập DS các lớp và lưu thông tin vào lophoc.inp !");
67         do{
68             printf("\nNhập số lượng sinh viên:");
69             scanf("%d", &n);
70             if(n<=0||n>100){
71                 printf("\nSai! Số lượng sinh viên phải nằm trong khoảng --- n>0||n>100-->nам on thu lai. ");
72             }
73             if(n<=0||n>100)
74                 nhapSV(n,a);
75             printf("\nBạn đã nhập thành công!");
76             daNhap = true;
77             printf("\nBạn phím bat kỵ để tiếp tục!");
78             getch();
79         }while(!daNhap);
80     case 2:
81         if(daNhap){
82             printf("\nBạn đã chọn xuất DS các lớp(file và xuất ra màn hình)! ");
83             xuatSV_file(f1,n,a);
84             printf("-----DAH SACH LOP DUOC XUAT VAO FILE: \n");
85             xuatSV_file(f1,n,a,f1);
86         }
87     case 3:
88         if(daNhap){
89             printf("\nBạn đã chọn (b)In danh sách lop co tren 5 hoc sinh gioi!");
90             //--In danh sach lop co tren 5 hoc sinh gioi
91             printf("\n-----Danh sach cac lop co nhieu hon 5 hoc sinh gioi: \n");
92             xuat_hsg(n,a);
93         }
94         else{
95             printf("\nNhập DS SV truoc!!!!!");
96         }
97         getch();
98     case 4:
99         if(daNhap){
100            printf("\nBạn đã chọn (c)Tim lop co nhieu hoc sinh nhat(de quy)!");
101            c=<--Tim lop co nhieu hoc sinh nhat(de quy)
102            int num_c = max_hs_dequy(n,a);
103            printf("\n-----Danh sach cac lop co nhieu hoc sinh nhat: ");
104            for(int i=0;i<n;i++){
105                if(a[i].siso==num_c){
106                    xuat(a[i]);
107                }
108            }
109        }
110        else{
111            printf("\nNhập DS SV truoc!!!!!");
112        }
113        getch();
114    }
115 }

```



The screenshot shows two instances of the Dev-C++ IDE running side-by-side. Both instances are displaying the same C++ source code for 'bai7.cpp'. The code is a menu-driven program that performs various operations on arrays, including sorting and searching. It includes functions for inputting and displaying array data. The IDE interface includes a toolbar, menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a tab bar (bai7.cpp, globals), status bar (Line: 4 Cob: 16 Sel: 0 Lines: 413 Length: 12495 Insert Done parsing in 0 seconds), and a system tray at the bottom right showing '30°C Light rain' and the date '20-Jul-21'.

```
157     getch();
158     break;
159     case 5:
160         if(daNhap){
161             printf("\nBan da chon (F)Xoa khoi danh sach cac lop co so hoc sinh yeu nhieu hon 1/2 si so!");
162             f=(xoa_khoi_danh_sach_cac_lop_co_so_hoc_sinh_yeu_nhieu_hon_1/2_si_so());
163             printf("\n---(F)Danh sach sinh vien sau khi xoa(so hs yeu > 1/2 si so): ");
164             xoa_ds(n,a);
165             xuatSV(n,a);
166         }
167         else{
168             printf("\nNhap DS SV truoc!!!!");
169         }
170         printf("\nBan phim bat ky de tiep tuc!");
171         getch();
172         break;
173     case 6:
174         if(daNhap){
175             rewind(f1);
176             printf("\nBan da chon (g) Sap xep danh sach cac lop tang theo si so bang quick sort va luu vao lophoc.out");
177             g=(sap_xep_danh_sach_cac_lop_tang_dan_theo_si_so_bang_quick_sort_va_luu_vao_lophoc.out);
178             fprintf(f2, "\nTinh tac xung SAP XEP LOP DUC XUAT VAO FILE:\n");
179             printf("\nBan da chon (g) Sap xep danh sach cac lop tang dan theo si so (quick sort): ");
180             fprintf(f2, "\n---(g)Danh sach cac lop tang dan theo si so (quick sort): ");
181             sap_xep_tang_siso(n,a,f2);
182         }
183         else{
184             printf("\nNhap DS SV truoc!!!!");
185         }
186         printf("\nBan phim bat ky de tiep tuc!");
187         getch();
188         break;
189     case 7:
190         fclose(f1);
191         fclose(f2);
192         printf("\nBan da chon thoat chuong trinh!");
193         getch();
194         return 0;
195     default:
196         printf("\nKhong co chuc nang nay!");
197         printf("\nBan phim bat ky de tiep tuc!");
198         getch();
199         break;
200     }
201 }
202 }

```

```
128     getch();
129     break;
130
131     case 5:
132         if(daNhap){
133             printf("\nBan da chon (d)Tim lop co so luong hoc sinh dat loni goi nhieu nhat!");
134             d=tim_lop_co_so_luong_hoc_sinh_dat_loui_goi_nhieu_nhat();
135             num_d = hsg_nhieu_nhat(n,a);
136             printf("\n(n-->) Danh sach cac lop co nhieu hoc sinh goi nhieu: ");
137             for(int i=0;i<n;i++){
138                 if(a[i].socong_hs_gioi==num_d){
139                     xuat(a[i]);
140                 }
141             }
142         }
143         else{
144             printf("\nNhap DS SV truoc!!!!");
145         }
146         printf("\nBan phim bat ky de tiep tuc!");
147         getch();
148         break;
149     case 6:
150         if(daNhap){
151             printf("\nBan da chon (e)Nhap ten giao vien cho biet thong tin lop gia vien phu trach(neu co)! ");
152             e=(nhap_ten_giao_vien_cho_biet_thong_tin_lop_gia_vien_phu_trach(neu_co));
153             char num_e[100];
154             printf("\n(n-->)Nhap ten giao vien can tim: ");
155             fflush(stdin);
156             gets(num_e);
157             tim_gv(num_a,num_e);
158         }
159         else{
160             printf("\nNhap DS SV truoc!!!!");
161         }
162         printf("\nBan phim bat ky de tiep tuc!");
163         getch();
164         break;
165     case 7:
166         if(daNhap){
167             printf("\nBan da chon (F)Xoa khoi danh sach cac lop co so hoc sinh yeu nhieu hon 1/2 si so!");
168             f=(xoa_khoi_danh_sach_cac_lop_co_so_hoc_sinh_yeu_nhieu_hon_1/2_si_so());
169             printf("\n---(F)Danh sach sinh vien sau khi xoa(so hs yeu > 1/2 si so): ");
170             xoa_ds(n,a);
171             xuatSV(n,a);
172         }
173         else{
174             printf("\nNhap DS SV truoc!!!!");
175         }
176     }
177 }
```

```

262 }
263
264 //nhap_t
265 void nhap (sv&x){
266     printf("\nNhap ten lop: ");
267     fflush(stdin);
268     gets(x.tenlop);
269     printf("nhap so ten gvcn: ");
270     scanf("%d", &x.siso);
271     printf("\nNhap ho va ten gvcn: ");
272     fflush(stdin);
273     gets(x.ten_gvcn);
274     printf("\nNhap so luong hoc sinh gioi: ");
275     scanf("%d", &x.soluong_hs_gioi);
276     printf("\nNhap so luong hoc sinh kha: ");
277     scanf("%d", &x.soluong_hs_kha);
278     printf("\nNhap so luong hoc sinh trung binh: ");
279     scanf("%d", &x.soluong_hs_trungbinh);
280     printf("\nNhap so luong hoc sinh yeu: ");
281     scanf("%d", &x.soluong_hs_yeu);
282 }
283 //nhap_mien
284 void nhapMSV(int n, sv a[]){
285     for (int i=0; i<n;i++){
286         printf("\n");
287         printf("Nhap sinh vien thu %d:",i+1);
288         nhap(a[i]);
289     }
290 }
291 //xuat_t
292 void xuat(sv x){
293     printf("\nTen lop: %s", x.tenlop);
294     printf("\nSo sinh vien: %d", x.siso);
295     printf("\nHo va ten gvcn: %s", x.ten_gvcn);
296     printf("\nSo luong hoc sinh gioi: %d", x.soluong_hs_gioi);
297     printf("\nSo luong hoc sinh kha: %d", x.soluong_hs_kha);
298     printf("\nSo luong hoc sinh trung binh: %d", x.soluong_hs_trungbinh);
299     printf("\nSo luong hoc sinh yeu: %d", x.soluong_hs_yeu);
300 }
301 //xuat_mieu
302 void xuatSV(int n, sv a[]){
303     for (int i=0; i<n;i++){
304         printf("\n---So thu: %d---",i+1);
305         xuat(a[i]);
306     }
307 }
308 //xuat_file
309 void xuat_file_f1(sv x,FILE *f1){
310     fprintf(f1,"%Ten lop: %s", x.tenlop);
311     fprintf(f1,"%So sinh vien: %d", x.siso);
312     fprintf(f1,"%Ho va ten gvcn: %s", x.ten_gvcn);
313     fprintf(f1,"%So luong hoc sinh gioi: %d", x.soluong_hs_gioi);
314     fprintf(f1,"%So luong hoc sinh kha: %d", x.soluong_hs_kha);
315     fprintf(f1,"%So luong hoc sinh trung binh: %d", x.soluong_hs_trungbinh);
316     fprintf(f1,"%So luong hoc sinh yeu: %d", x.soluong_hs_yeu);
317 }
318 void xuat_file_f2(sv x,FILE *f2){
319     fprintf(f2,"%Ten lop: %s", x.tenlop);
320     fprintf(f2,"%So sinh vien: %d", x.siso);
321     fprintf(f2,"%Ho va ten gvcn: %s", x.ten_gvcn);
322     fprintf(f2,"%So luong hoc sinh gioi: %d", x.soluong_hs_gioi);
323     fprintf(f2,"%So luong hoc sinh kha: %d", x.soluong_hs_kha);
324     fprintf(f2,"%So luong hoc sinh trung binh: %d", x.soluong_hs_trungbinh);
325     fprintf(f2,"%So luong hoc sinh yeu: %d", x.soluong_hs_yeu);
326 }
327 //xuat_sv_file
328 void xuatSV_file_f1(int n, sv a[],FILE *f1){
329     for (int i=0; i<n;i++){
330         fprintf(f1,"%\n---So thu %d---",i+1);
331         xuat_file_f1(a[i],f1);
332         printf("\n");
333     }
334 }
335 void xuatSV_file_f2(int n, sv a[],FILE *f2){
336     for (int i=0; i<n;i++){
337         fprintf(f2,"%\n---So thu %d---",i+1);
338         xuat_file_f2(a[i],f2);
339         fprintf(f2,"%\n");
340     }
341 }
342 /////
343 void xuat_hsg(int n, sv a[]){
344     for (int i=0; i<n;i++){
345         if(a[i].soluong_hs_gioi > 5){
346             xuat(a[i]);
347         }
348     }
349 }

```

```

283 }
284 //ds 5 hs gioi
285 void xuat_hs(int n, sv a[]){
286     for(int i=0; i<n;i++){
287         if(a[i].soluong_hs_giol > 5){
288             xuat(a[i]);
289         }
290     }
291 }
292 //c///
293 //lop max hs what_de quy
294 int max_hs_dequy(int n, sv a[]){
295     if(n == 1){
296         return a[0].siso;
297     }
298     if(n==0){
299         return 0;
300     }
301     if(max_hs_dequy(n-1,a)>(n-1).siso){
302         return max_hs_dequy(n-1, a);
303     }else{
304         printf("%d\n", n-1);
305         return a[n-1].siso;
306     }
307 }
308 //d/
309 //lop to hsg nhieu nhat
310 int hsg_nhieu_nhhat(int n, sv a[]){
311     int max=-1;
312     for(int i=0; i<n;i++){
313         if(a[i].soluong_hs_giol > max){
314             max = a[i].soluong_hs_giol;
315         }
316     }
317     return max;
318 }
319 //t/
320 //ham ten gv tim thong tin
321 void tim_gv(int n, sv a[], char ten_gv[]){
322     for(int i=0; i<n;i++){
323         if(strcmp(a[i].hoten_gvcn, ten_gv) == 0){
324             xuat(a[i]);
325         }
326     }
327 }
328 }

329 } //fff
330 //xoa ds if (hs.yeuu /2 h )
331 void xoa_ds(int &n, sv a[]){
332     int flag=0;
333     for(int i=0; i<n;i++){
334         if(a[i].soluong_hs_yeuu > (a[i].siso / 2)){
335             flag = 1;
336             for(int j=i; j<n;j++){
337                 a[j]=a[j+1];
338             }
339             printf("\nDa xoa loi %s ", a[j].tenlo);
340             break;
341         }
342     }
343     if(flag == 0){
344         printf("\nKhong co loi can xoa");
345     }
346 }
347 //g
348 //sap xep tang dan theo so size = quick sort.
349 void swap(int &a, int &b)
350 {
351     int t = a;
352     a = b;
353     b = t;
354 }
355 int partition(int a[], int left, int right)
356 {
357     int i = left;
358     int j = right - 1;
359     int pivot = a[right];
360     while (true)
361     {
362         while (pivot > a[i] && i <= j)
363             i++;
364         while (pivot < a[j] && i <= j)
365             j--;
366         if (i >= j)
367         {
368             break;
369         }
370         else
371         {
372             swap(a[i], a[j]);
373         }
374     }
375 }

```

```

355     }
356     int partition(int a[], int left, int right)
357     {
358         int i = left;
359         int j = right - i;
360         int pivot = a[right];
361         while (true)
362         {
363             while (pivot > a[i] && i <= j)
364                 i++;
365             while (pivot < a[j] && i <= j)
366                 j--;
367             if (i >= j)
368             {
369                 break;
370             }
371             else
372             {
373                 swap(a[i], a[j]);
374                 i++;
375                 j--;
376             }
377         }
378         swap(a[i], a[right]);
379         return i;
380     }
381     void quicksort(int arr[], int left, int right)
382     {
383         if (left <= right)
384         {
385             int pi = partition(arr, left, right);
386             quicksort(arr, left, pi - 1);
387             quicksort(arr, pi + 1, right);
388         }
389     }
390     int tim_kiem(int n, sv a[], int b,FILE *f2){
391         sv c[100];
392         for(int i=0; i<n;i++){
393             if(b==a[i].siso){
394                 xuat(a[i]);
395                 xuat_file_f2(a[i],f2);
396             }
397         }
398     }
399     void sap_xep_tang_siso(int n, sv a[], FILE *f2){
400         int b[100], k=0;
411     }

```

File Edit Search View Project Execute Tools AStyle Window Help

Compiler Resources Compile Log Debug Find Results

Line: 4 Col: 16 Sel: 0 Lines: 413 Length: 12495 Insert Done parsing in 0 seconds

800 AM 30°C Light rain 20-Jul-21

```

368     {
369         break;
370     }
371     else
372     {
373         swap(a[i], a[j]);
374         i++;
375         j--;
376     }
377     swap(a[i], a[right]);
378     return i;
379 }
380 void quicksort(int arr[], int left, int right)
381 {
382     if (left <= right)
383     {
384         int pi = partition(arr, left, right);
385         quicksort(arr, left, pi - 1);
386         quicksort(arr, pi + 1, right);
387     }
388 }
389 int tim_kiem(int n, sv a[], int b,FILE *f2){
390     sv c[100];
391     for(int i=0; i<n;i++){
392         if(b==a[i].siso){
393             xuat(a[i]);
394             xuat_file_f2(a[i],f2);
395         }
396     }
397 }
398 void sap_xep_tang_siso(int n, sv a[], FILE *f2){
399     int b[100];
400     for(int i=0; i<n;i++){
401         b[k++]=a[i].siso;
402     }
403     quicksort(b, 0, n-1);
404     for(int i=0; i<n;i++){
405         tim_kiem(n,a,b[i],f2);
406     }
407 }
408
409 }
410
411
412
413

```

File Edit Search View Project Execute Tools AStyle Window Help

Compiler Resources Compile Log Debug Find Results

Line: 4 Col: 16 Sel: 0 Lines: 413 Length: 12495 Insert Done parsing in 0 seconds

800 AM 30°C Light rain 20-Jul-21