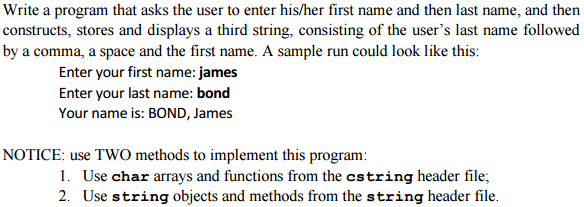
**Assignment 1**

**1301058**

**Zhang Junming**

**Exercise 1**

Question



Model Answer

Software Development Process

1. **Problem statement**

Use two ways ( “char arrays and functions from the cstring header file” and “string objects and methods from the string header file” ) to write a program that could ask user to enter their first and last name, then display their full name with last name in capital letter.

1. **Analysis**

Inputs:

1. First name and last name.

Outputs:

1. Full name with last name in capital letter.

Additional requirements or constraint

Using two ways: char arrays and string objects.

Output all information in 3rd string.

**3. Design**

Algorithm

First way (char arrays):

1. Adding “iostream” and “cstring” header files.
2. Using of the ste namespace.
3. Write main function.

(1) char Firstname[20] – to store the user’s first name.

(2) char Lastname[20] – to store the user’s last name.

(3) char Fullname[40] = {} – to store the user’s full name and initialize this array.

(4) int i = 0 – for setting loop to convert letters.

(5) int j = 0 – for connect first name and last name.

(6) Ask user to enter their first name.

(7) Read and store user’s first name in array Firstname.

(8) Ask user to enter their last name.

(9) Read and store user’s first name in array Lastname.

(10) Convert last name to capital letter using for loop.

<1> counting how many letters in array Lastname.

<2> setting up a for loop.

<3> judge each letter in array Lastname, if this letter is lowercase and then convert it to capital letter, if not, keep it. ( ACSLL: lowercase – 32 = capital letter)

<4> Store last name in array Fullname.

(11) Store “,” and “ “ in array Fullname.

(12) Record the next location of array Fullname in j.

(13) Store first name in array Fullname using for loop.

<1> counting how many letters in array Firstname.

<2> setting up a for loop.

<3> store first name in array Fullname.

(14) Display user’s full name.

(15) The return type of main() must always be an int.

Second way ( string ):

1. Adding “iostream” and “string” header files.
2. Using of the ste namespace.
3. Write main function.

(1)string Firstname – to store the user’s first name.

(2)string Lastname – to store the user’s last name.

(3)string Fullname – to store the user’s full name.

(4)string Other = “, “ – represents “, “ between first name and last name.

(5) int i = 0 – for setting loop to convert and store letters.

(6) Ask user to enter their first name.

(7) Read and store user’s first name in string Firstname.

(8) Ask user to enter their last name.

(9) Read and store user’s first name in string Lastname.

(10) Convert last name to capital letter using for loop.

<1> counting how many letters in string Lastname.

<2> setting up a for loop.

<3> judge each letter in string Lastname, if this letter is lowercase and then convert it to capital letter, if not, keep it. ( toupper() function in library string could convert lowercase to capital letter )

<4> Store last name in string Lastname.

(11) Connect string Lastname, Other and Firstname to string Fullname.

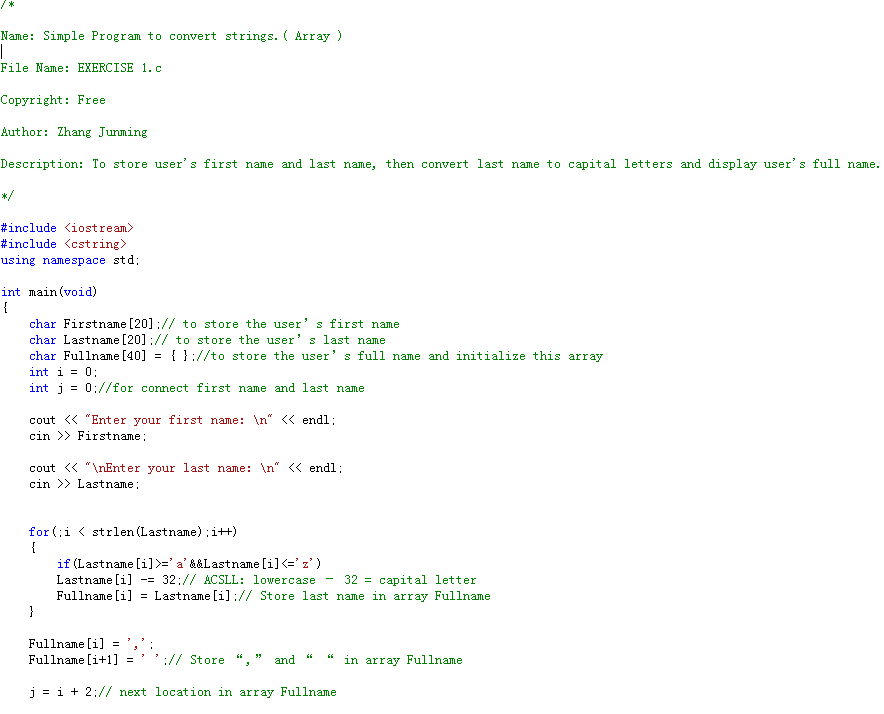
(12) Display user’s full name.

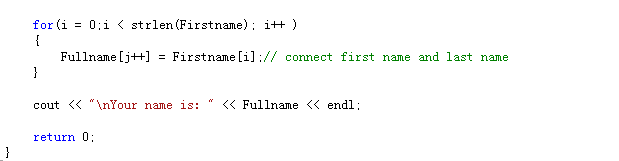
(13) The return type of main() must always be an int.

1. **Implementation**:

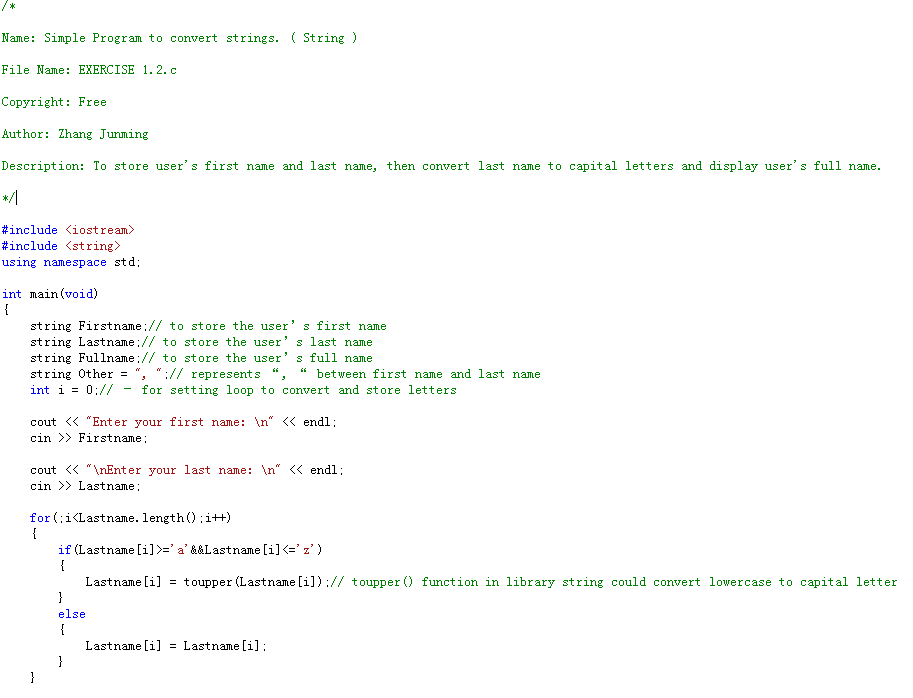
See the C code in file exercise1.c with comments.

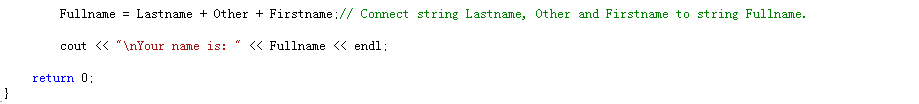
First way (char arrays):





Second way ( string ):

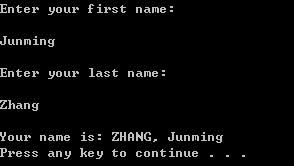




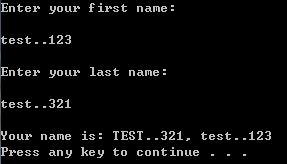
1. **Testing:**

First way (char arrays):

Normal test

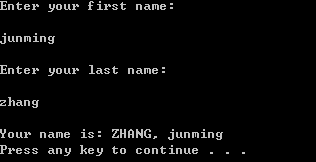


Only convert lowercase

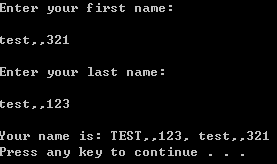


Second way ( string ):

Normal test

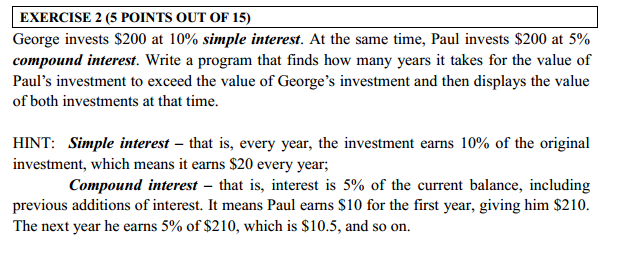


Only convert lowercase



**Exercise 2**

Question



Model Answer

Software Development Process

**1**. **Problem statement**

Write a program to calculate after how many years the investment of compound interest will exceed simple interest with the same invests and display their property at that time.

**2. Analysis**

Inputs:

The value of invests, compound and simple interest.

Outputs:

After how many years the investment of compound interest exceeds simple interest and their property.

Additional requirements or constraint

They begin to make money at the first year.

**3. Design**

Algorithm

1. Adding “iostream” header files.

2. Define simple interest is 0.1 and compound interest is 0.05.

3. Using of the ste namespace.

1. Write main function.

(1) int George\_invest = 200 – represents George’s investment is ＄200

(2) int i = 0 – represents time. (years)

(3) int j = 0 – to judge whether out the loop.

(4) int George\_Property = 200 – to store George’s main property and at the first time George has ＄200.

(5) float Paul\_Property = 0 – to store Paul’s main property and at the first time Paul has ＄200.

(6) Display this program’s function

(7) Setting up a loop using do-while loop.

<1> calculates and input George’s simple interest function.

<2> calculates and input Paul’s compound interest function.

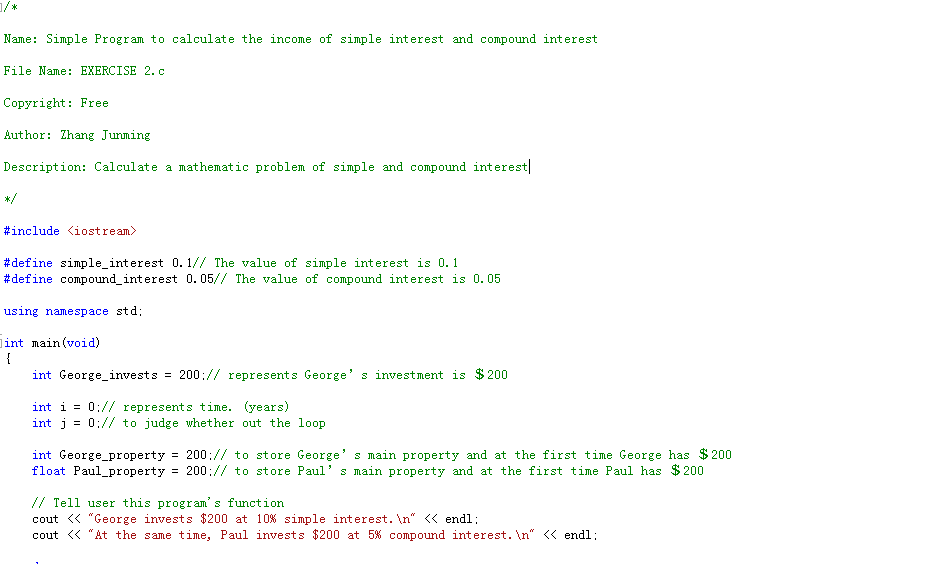
<3> judge Paul and George’s property, if Paul’s property exceed George’s, out the loop, else i+1.

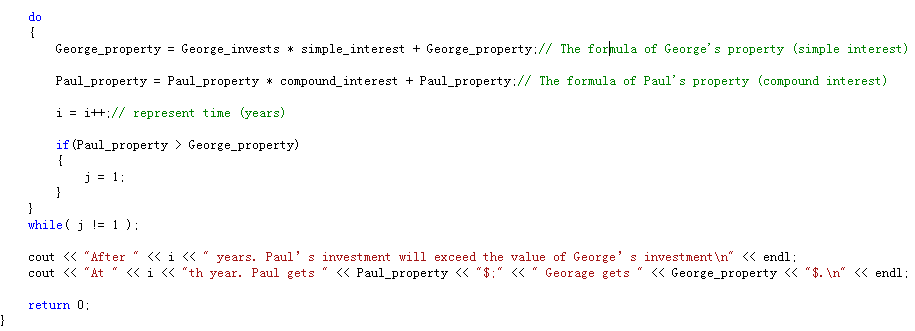
(8) Display the result.

(9) The return type of main() must always be an int.

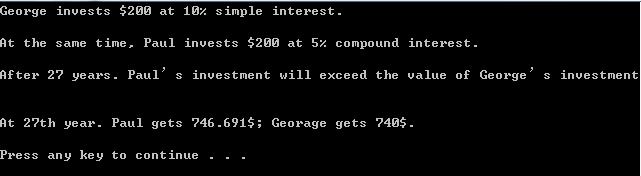
1. **Implementation**:

See the C code in file exercise1.c with comments.



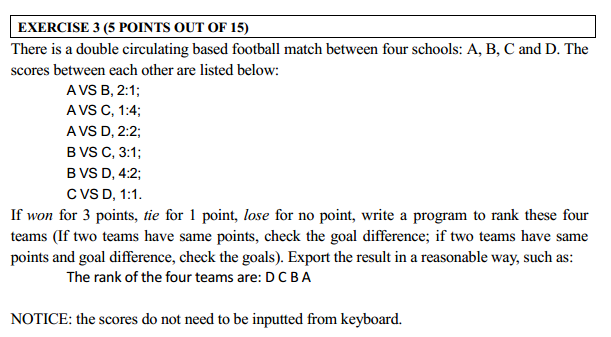


1. **Testing:**



**Exercise 3**

Question



Model Answer

Software Development Process

**1**. **Problem statement**

Write a program to rank 4 teams, based on their rank point, goal difference and goals.

1. **Analysis**

Inputs:

The results of 6 matches.

Outputs:

The rank of 4 teams.

Additional requirements or constraint

The scores do not need to be inputted from keyboard.

**3. Design**

Algorithm

1. Adding “iostream” header files.

2. Using of the ste namespace.

3. Write main function.

(1) int a,b,c,d = 4 – for sorting

(2) Display the results of 6 matches

(3) int A,B,C,D\_Rank = 0 – to store the rank points of 4 teams

(4) int A,B,C,D\_GD – to store the goal difference of 4 teams.

(5) int A,B,C,D\_goal – to store the goals of 4 teams.

(6) int A\_B = 2 – represents the goals which A got in the match of team A and B.

(7) int A\_C = 1 – represents the goals which A got in the match of team A and C.

(8) int A\_D = 2 – represents the goals which A got in the match of team A and D.

(9) int B\_A = 1 – represents the goals which B got in the match of team A and B.

(10) int B\_C = 3 – represents the goals which B got in the match of team C and B.

(11) int B\_D = 4 – represents the goals which B got in the match of team D and B.

(12) int C\_A = 4 – represents the goals which C got in the match of team C and A.

(12) int C\_B = 2 – represents the goals which C got in the match of team C and B.

(13) int C\_D = 1 – represents the goals which C got in the match of team C and D.

(14) int D\_A = 2 – represents the goals which D got in the match of team D and A.

(15) int D\_B = 2 – represents the goals which D got in the match of team D and B.

(16) int D\_C = 1 – represents the goals which D got in the match of team D and C.

(17)Calculate the rank points of each team.

<1> if A won B, A get 3 points; if A tie B, A and B get 1 points;

<2> if A won C, A get 3 points; if A tie C, A and C get 1 points;

<3> if A won D, A get 3 points; if A tie D, A and D get 1 points;

<4> if B won A, B get 3 points;

<5> if B won C, B get 3 points; if B tie C, B and C get 1 points;

<6> if B won D, B get 3 points; if B tie D, B and D get 1 points;

<7> if C won A, C get 3 points;

<8> if C won B, C get 3 points;

<9> if C won D, C get 3 points; if C tie D, C and D get 1 points;

<10> if D won A, D get 3 points;

<11> if D won B, D get 3 points;

<10> if D won C, D get 3 points;

(18) Calculate the goals of each team.

<1> the goals of team A equal to the sum of A\_B,C,D.

<2> the goals of team B equal to the sum of B\_A,C,D.

<3> the goals of team C equal to the sum of C\_B,A,D.

<4> the goals of team D equal to the sum of D\_B,C,A.

(19) Calculate the goal difference of each team.

<1> the goal difference of team A equal to the difference of goals of A and B,C,D\_A.

<2> the goal difference of team B equal to the difference of goals of B and A,C,D\_B.

<3> the goal difference of team C equal to the difference of goals of C and B,A,D\_C.

<4> the goal difference of team D equal to the difference of goals of D and B,C,A\_D.

(20) Sort.

<1> at the beginning, all rank of 4 team are 4th

<2> the condition of A>B, A=B and A<B. if the rank points of team A bigger than team B, the rank of A will improve. Else if the rank points are same, then compare their goal difference, if A bigger than B, the rank of A will improve. Else if the goal difference are same, then compare their goals, if A bigger than B, the rank of A will improve. On the other hand, if A smaller than B, then the rank of B will improve.

<3> the same as last step, compare the condition of A>C, A=C and A<C; A>D, A=D and A<D; B>C, B=C and B<C; B>D, B=D and B<D; C>D, C=D and C<D;

<4> Now, the value of rank will store in a,b,c,d.

(21) Display the result

<1> judge the value of a,b,c,d.

<2> if a = 1, display team A is first, and then display the rank points, goal difference and goals of team A on screen.

<3> if a = 2, display team A is second, and then display the rank points, goal difference and goals of team A on screen.

<4> if a = 3, display team A is third, and then display the rank points, goal difference and goals of team A on screen.

<5> if a = 4, display team A is fourth, and then display the rank points, goal difference and goals of team A on screen.

<6> the same as last step, display the rank result of all teams.

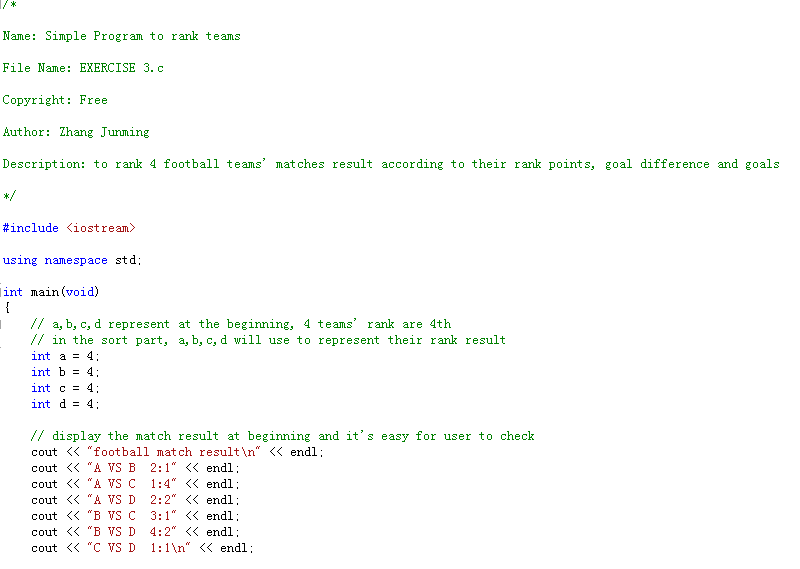
(22) The return type of main() must always be an int.

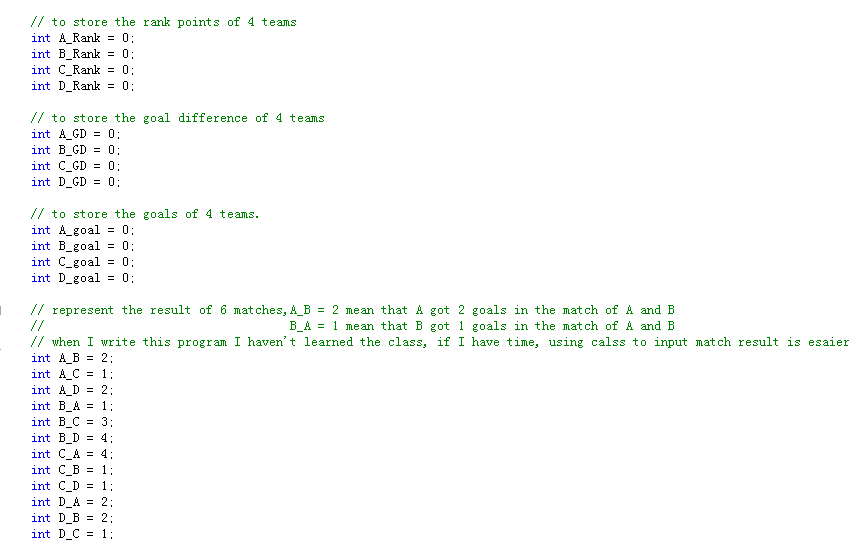
**IMPROVE**

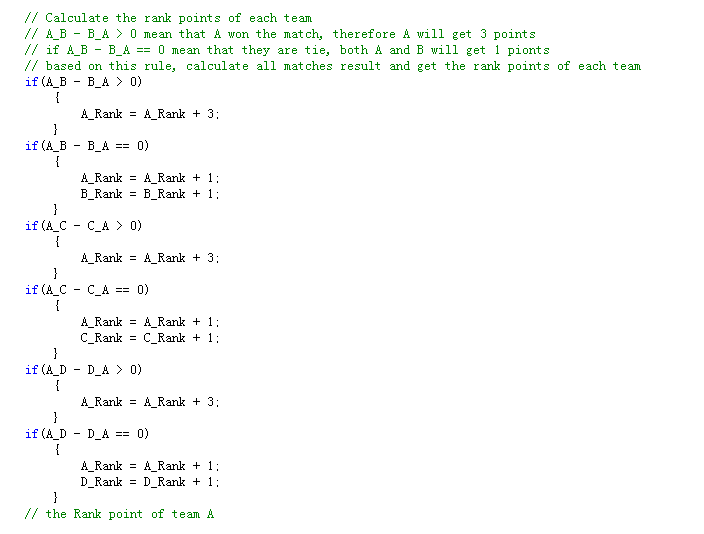
When I write this program, I haven’t learned the lecture 2, therefore I didn’t know the class. If I use class to classify, this program will more easy to write, and I could summarize the parts of sort and display result to an independent function, and then my program will more concise.

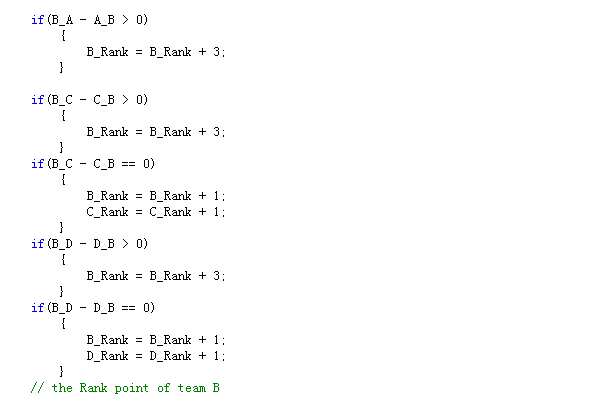
1. **Implementation**:

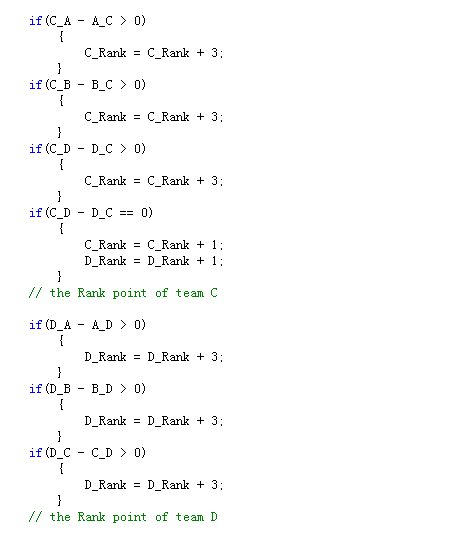
See the C code in file exercise1.c with comments.

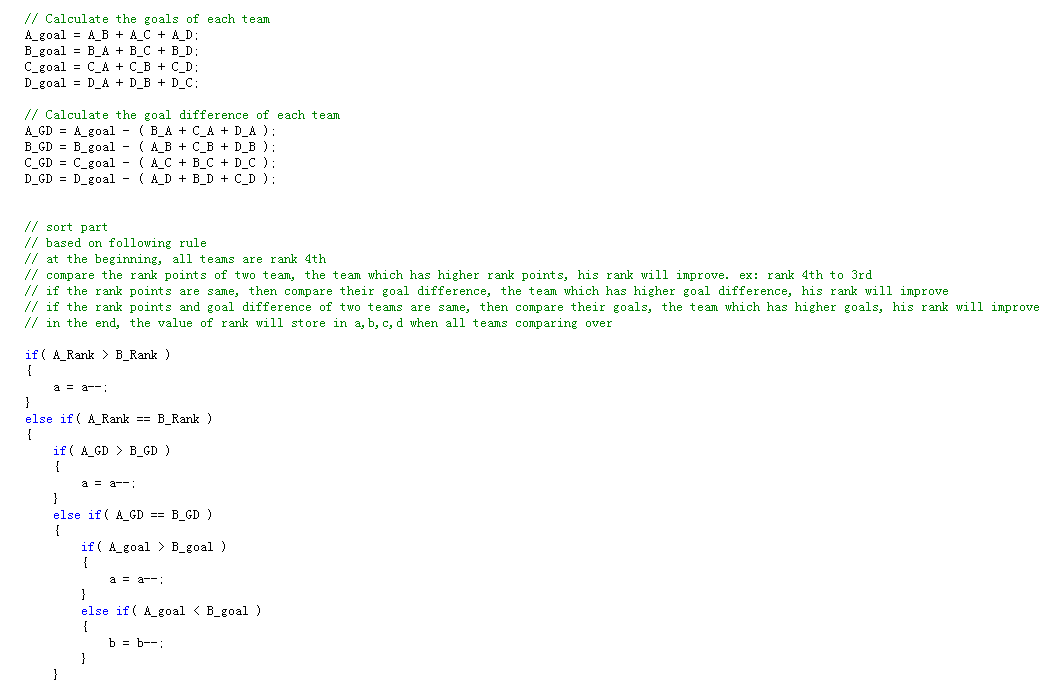


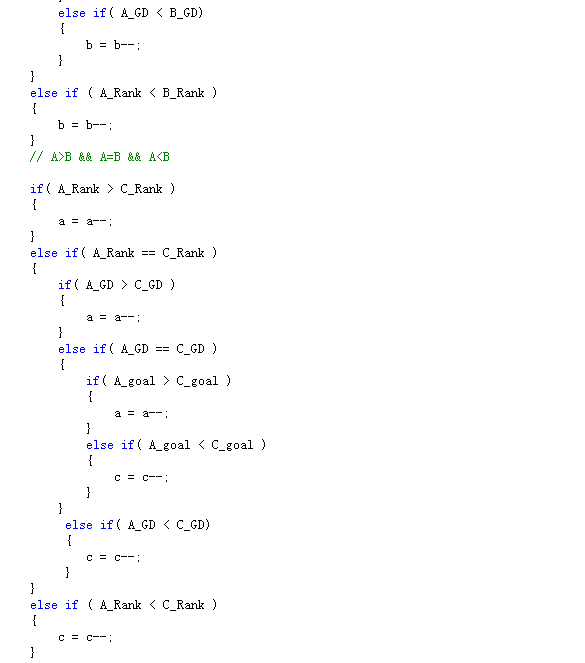


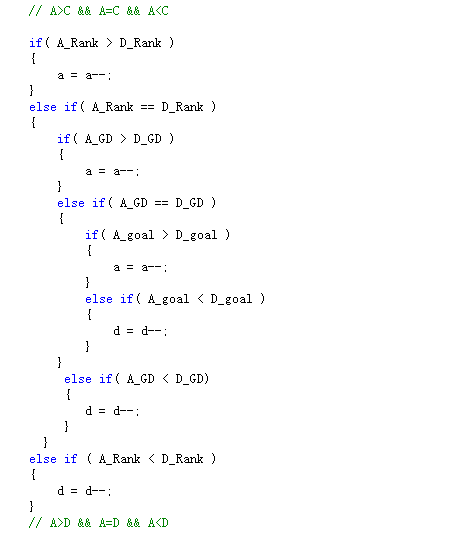


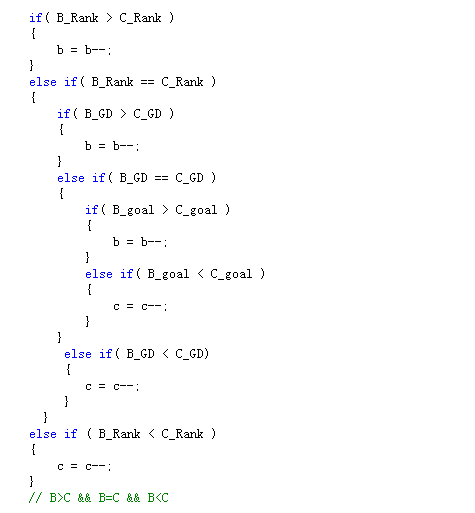


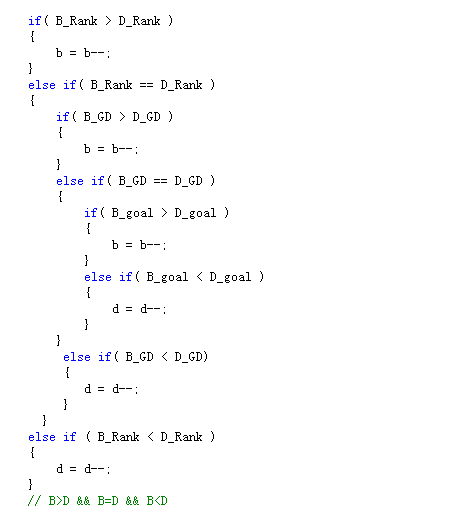


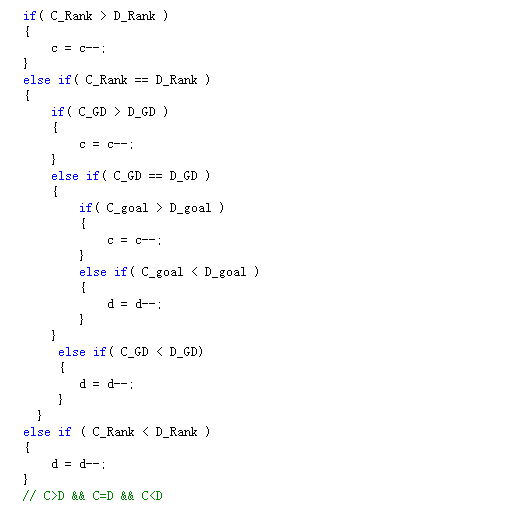


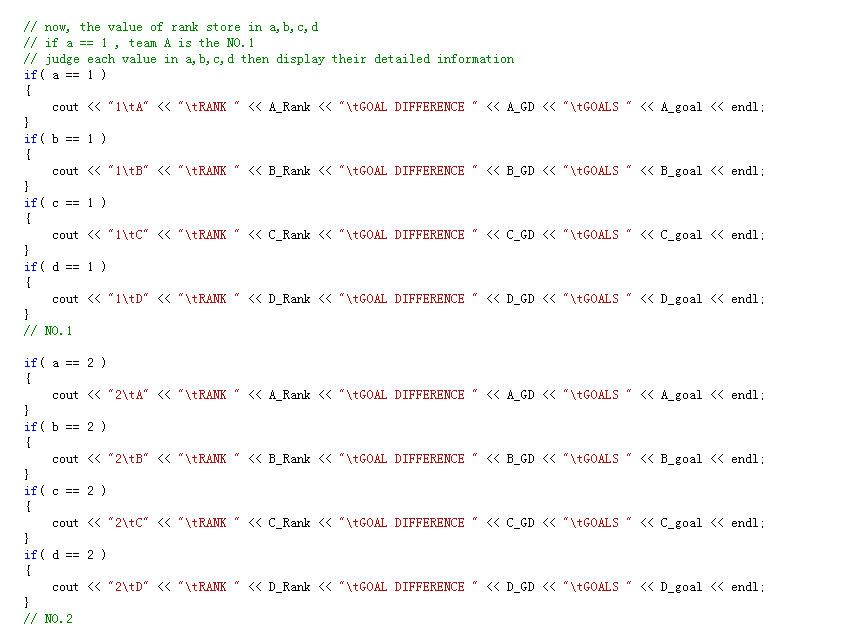


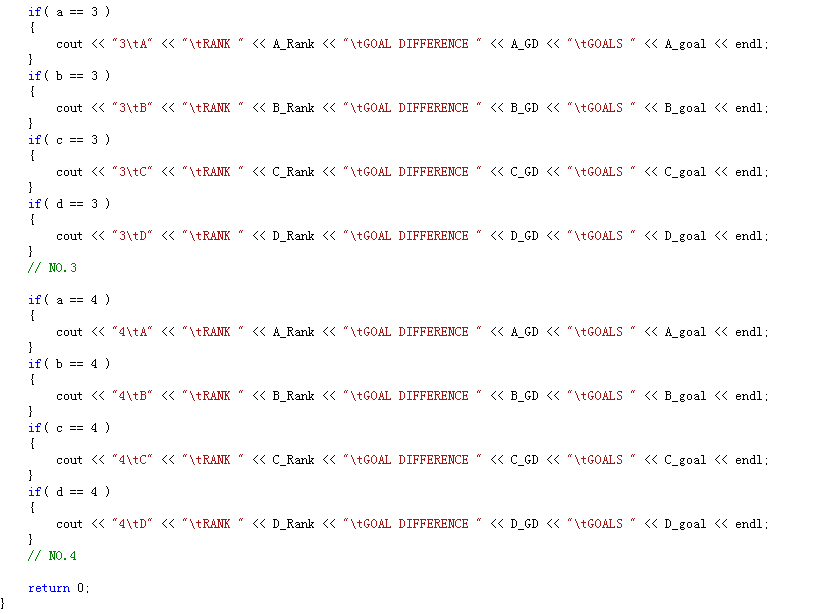












1. **Testing:**

