

Assignment 1

1301058

Zhang Junming

Exercise 1

Question

Write a program that asks the user to enter his/her first name and then last name, and then constructs, stores and displays a third string, consisting of the user's last name followed by a comma, a space and the first name. A sample run could look like this:

Enter your first name: **james**

Enter your last name: **bond**

Your name is: BOND, James

NOTICE: use TWO methods to implement this program:

1. Use **char** arrays and functions from the **cstring** header file;
2. Use **string** objects and methods from the **string** header file.

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.

2. 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 std 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. (ASCII: 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 std 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`.

4. Implementation:

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

First way (char arrays):

```

/*
Name: Simple Program to convert strings. ( Array )
|
File Name: EXERCISE 1.c

Copyright: Free

Author: Zhang Jurming

Description: To store user's first name and last name, then convert last name to capital letters and display user's full name.

*/

#include <iostream>
#include <cstring>
using namespace std;

int main(void)
{
    char Firstname[20]; // to store the user's first name
    char Lastname[20]; // to store the user's last name
    char Fullname[40] = { }; // to store the user's full name and initialize this array
    int i = 0;
    int j = 0; // for connect first name and last name

    cout << "Enter your first name: \n" << endl;
    cin >> Firstname;

    cout << "\nEnter your last name: \n" << endl;
    cin >> Lastname;

    for(; i < strlen(Lastname); i++)
    {
        if(Lastname[i] >= 'a' && Lastname[i] <= 'z')
            Lastname[i] -= 32; // ASCII: lowercase - 32 = capital letter
        Fullname[i] = Lastname[i]; // Store last name in array Fullname
    }

    Fullname[i] = ',';
    Fullname[i+1] = ' '; // Store ",", " " and " " in array Fullname

    j = i + 2; // next location in array Fullname

    for(i = 0; i < strlen(Firstname); i++)
    {
        Fullname[j++] = Firstname[i]; // connect first name and last name
    }

    cout << "\nYour name is: " << Fullname << endl;

    return 0;
}

```

Second way (string):

```
/*  
  
Name: Simple Program to convert strings. ( String )  
  
File Name: EXERCISE 1.2.c  
  
Copyright: Free  
  
Author: Zhang Jurming  
  
Description: To store user's first name and last name, then convert last name to capital letters and display user's full name.  
  
*/  
  
#include <iostream>  
#include <string>  
using namespace std;  
  
int main(void)  
{  
    string Firstname;// to store the user's first name  
    string Lastname;// to store the user's last name  
    string Fullname;// to store the user's full name  
    string Other = ", ";// represents ", " between first name and last name  
    int i = 0;// - for setting loop to convert and store letters  
  
    cout << "Enter your first name: \n" << endl;  
    cin >> Firstname;  
  
    cout << "\nEnter your last name: \n" << endl;  
    cin >> Lastname;  
  
    for(;i<Lastname.length();i++)  
    {  
        if(Lastname[i]>='a' && Lastname[i]<='z')  
        {  
            Lastname[i] = toupper(Lastname[i]);// toupper() function in library string could convert lowercase to capital letter  
        }  
        else  
        {  
            Lastname[i] = Lastname[i];  
        }  
    }  
  
    Fullname = Lastname + Other + Firstname;// Connect string Lastname, Other and Firstname to string Fullname.  
  
    cout << "\nYour name is: " << Fullname << endl;  
  
    return 0;  
}
```

5. Testing:

First way (char arrays):

Normal test

```
Enter your first name:
Junming
Enter your last name:
Zhang
Your name is: ZHANG, Junming
Press any key to continue . . .
```

Only convert lowercase

```
Enter your first name:
test..123
Enter your last name:
test..321
Your name is: TEST..321, test..123
Press any key to continue . . .
```

Second way (string):

Normal test

```
Enter your first name:
junming
Enter your last name:
zhang
Your name is: ZHANG, junming
Press any key to continue . . .
```

Only convert lowercase

```
Enter your first name:
test,,321

Enter your last name:
test,,123

Your name is: TEST,,123, test,,321
Press any key to continue . . .
```

Exercise 2

Question

EXERCISE 2 (5 POINTS OUT OF 15)

George invests \$200 at 10% *simple interest*. At the same time, Paul invests \$200 at 5% *compound interest*. Write a program that finds how many years it takes for the value of Paul's investment to exceed the value of George's investment and then displays the value of both investments at that time.

HINT: *Simple interest* – that is, every year, the investment earns 10% of the original investment, which means it earns \$20 every year;

Compound interest – that is, interest is 5% of the current balance, including previous additions of interest. It means Paul earns \$10 for the first year, giving him \$210. The next year he earns 5% of \$210, which is \$10.5, and so on.

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 std namespace.
4. 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.

6. Implementation:

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

```

/*
Name: Simple Program to calculate the income of simple interest and compound interest
File Name: EXERCISE 2.c
Copyright: Free
Author: Zhang Junming
Description: Calculate a mathematic problem of simple and compound interest
*/

#include <iostream>

#define simple_interest 0.1// The value of simple interest is 0.1
#define compound_interest 0.05// The value of compound interest is 0.05

using namespace std;

int main(void)
{
    int George_invests = 200;// represents George' s investment is $200

    int i = 0;// represents time. (years)
    int j = 0;// to judge whether out the loop

    int George_property = 200;// to store George' s main property and at the first time George has $200
    float Paul_property = 200;// to store Paul' s main property and at the first time Paul has $200

    // Tell user this program's function
    cout << "George invests $200 at 10% simple interest.\n" << endl;
    cout << "At the same time, Paul invests $200 at 5% compound interest.\n" << endl;
    .

```

```

do
{
    George_property = George_invests * simple_interest + George_property; // The formula of George's property (simple interest)

    Paul_property = Paul_property * compound_interest + Paul_property; // The formula of Paul's property (compound interest)

    i = i++; // represent time (years)

    if(Paul_property > George_property)
    {
        j = 1;
    }
}
while( j != 1 );

cout << "After " << i << " years. Paul's investment will exceed the value of George's investment\n" << endl;
cout << "At " << i << "th year. Paul gets " << Paul_property << "$;" << " George gets " << George_property << "$.\n" << endl;

return 0;
}

```

7. Testing:

```

George invests $200 at 10% simple interest.

At the same time, Paul invests $200 at 5% compound interest.

After 27 years. Paul's investment will exceed the value of George's investment

At 27th year. Paul gets 746.691$; Georage gets 740$.

Press any key to continue . . .

```

Exercise 3

Question

EXERCISE 3 (5 POINTS OUT OF 15)

There is a double circulating based football match between four schools: A, B, C and D. The scores between each other are listed below:

A VS B, 2:1;

A VS C, 1:4;

A VS D, 2:2;

B VS C, 3:1;

B VS D, 4:2;

C VS D, 1:1.

If *won* for 3 points, *tie* for 1 point, *lose* for no point, write a program to rank these four teams (If two teams have same points, check the goal difference; if two teams have same points and goal difference, check the goals). Export the result in a reasonable way, such as:

The rank of the four teams are: D C B A

NOTICE: the scores do not need to be inputted from keyboard.

Model Answer

Software Development Process

1. Problem statement

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

3. 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 std 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) $\text{int } B_D = 4$ – represents the goals which B got in the match of team D and B.

(12) $\text{int } C_A = 4$ – represents the goals which C got in the match of team C and A.

(12) $\text{int } C_B = 2$ – represents the goals which C got in the match of team C and B.

(13) $\text{int } C_D = 1$ – represents the goals which C got in the match of team C and D.

(14) $\text{int } D_A = 2$ – represents the goals which D got in the match of team D and A.

(15) $\text{int } D_B = 2$ – represents the goals which D got in the match of team D and B.

(16) $\text{int } D_C = 1$ – 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.

8. Implementation:

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

```
/*  
  
Name: Simple Program to rank teams  
  
File Name: EXERCISE 3.c  
  
Copyright: Free  
  
Author: Zhang Junming  
  
Description: to rank 4 football teams' matches result according to their rank points, goal difference and goals  
  
*/  
  
#include <iostream>  
  
using namespace std;  
  
int main(void)  
{  
    // a,b,c,d represent at the beginning, 4 teams' rank are 4th  
    // in the sort part, a,b,c,d will use to represent their rank result  
    int a = 4;  
    int b = 4;  
    int c = 4;  
    int d = 4;  
  
    // display the match result at beginning and it's easy for user to check  
    cout << "football match result\n" << endl;  
    cout << "A VS B  2:1" << endl;  
    cout << "A VS C  1:4" << endl;  
    cout << "A VS D  2:2" << endl;  
    cout << "B VS C  3:1" << endl;  
    cout << "B VS D  4:2" << endl;  
    cout << "C VS D  1:1\n" << endl;
```

```

// to store the rank points of 4 teams
int A_Rank = 0;
int B_Rank = 0;
int C_Rank = 0;
int D_Rank = 0;

// to store the goal difference of 4 teams
int A_GD = 0;
int B_GD = 0;
int C_GD = 0;
int D_GD = 0;

// to store the goals of 4 teams.
int A_goal = 0;
int B_goal = 0;
int C_goal = 0;
int D_goal = 0;

// represent the result of 6 matches,A_B = 2 mean that A got 2 goals in the match of A and B
//                                     B_A = 1 mean that B got 1 goals in the match of A and B
// when I write this program I haven't learned the class, if I have time, using calss to input match result is esaier
int A_B = 2;
int A_C = 1;
int A_D = 2;
int B_A = 1;
int B_C = 3;
int B_D = 4;
int C_A = 4;
int C_B = 1;
int C_D = 1;
int D_A = 2;
int D_B = 2;
int D_C = 1;

```

```

// Calculate the rank points of each team
// A_B - B_A > 0 mean that A won the match, therefore A will get 3 points
// if A_B - B_A == 0 mean that they are tie, both A and B will get 1 pionts
// based on this rule, calculate all matches result and get the rank points of each team
if(A_B - B_A > 0)
{
    A_Rank = A_Rank + 3;
}
if(A_B - B_A == 0)
{
    A_Rank = A_Rank + 1;
    B_Rank = B_Rank + 1;
}
if(A_C - C_A > 0)
{
    A_Rank = A_Rank + 3;
}
if(A_C - C_A == 0)
{
    A_Rank = A_Rank + 1;
    C_Rank = C_Rank + 1;
}
if(A_D - D_A > 0)
{
    A_Rank = A_Rank + 3;
}
if(A_D - D_A == 0)
{
    A_Rank = A_Rank + 1;
    D_Rank = D_Rank + 1;
}
// the Rank point of team A

```

```
if (B_A - A_B > 0)
{
    B_Rank = B_Rank + 3;
}

if (B_C - C_B > 0)
{
    B_Rank = B_Rank + 3;
}
if (B_C - C_B == 0)
{
    B_Rank = B_Rank + 1;
    C_Rank = C_Rank + 1;
}
if (B_D - D_B > 0)
{
    B_Rank = B_Rank + 3;
}
if (B_D - D_B == 0)
{
    B_Rank = B_Rank + 1;
    D_Rank = D_Rank + 1;
}
// the Rank point of team B
```

```

if(C_A - A_C > 0)
{
    C_Rank = C_Rank + 3;
}
if(C_B - B_C > 0)
{
    C_Rank = C_Rank + 3;
}
if(C_D - D_C > 0)
{
    C_Rank = C_Rank + 3;
}
if(C_D - D_C == 0)
{
    C_Rank = C_Rank + 1;
    D_Rank = D_Rank + 1;
}
// the Rank point of team C

if(D_A - A_D > 0)
{
    D_Rank = D_Rank + 3;
}
if(D_B - B_D > 0)
{
    D_Rank = D_Rank + 3;
}
if(D_C - C_D > 0)
{
    D_Rank = D_Rank + 3;
}
// the Rank point of team D

```

```

// Calculate the goals of each team
A_goal = A_B + A_C + A_D;
B_goal = B_A + B_C + B_D;
C_goal = C_A + C_B + C_D;
D_goal = D_A + D_B + D_C;

// Calculate the goal difference of each team
A_GD = A_goal - ( B_A + C_A + D_A );
B_GD = B_goal - ( A_B + C_B + D_B );
C_GD = C_goal - ( A_C + B_C + D_C );
D_GD = D_goal - ( A_D + B_D + C_D );

// sort part
// based on following rule
// at the beginning, all teams are rank 4th
// compare the rank points of two team, the team which has higher rank points, his rank will improve. ex: rank 4th to 3rd
// if the rank points are same, then compare their goal difference, the team which has higher goal difference, his rank will improve
// if the rank points and goal difference of two teams are same, then compare their goals, the team which has higher goals, his rank will improve
// in the end, the value of rank will store in a,b,c,d when all teams comparing over

if( A_Rank > B_Rank )
{
    a = a--;
}
else if( A_Rank == B_Rank )
{
    if( A_GD > B_GD )
    {
        a = a--;
    }
    else if( A_GD == B_GD )
    {
        if( A_goal > B_goal )
        {
            a = a--;
        }
        else if( A_goal < B_goal )
        {
            b = b--;
        }
    }
}

```

```

        else if( A_GD < B_GD)
        {
            b = b--;
        }
    }
    else if ( A_Rank < B_Rank )
    {
        b = b--;
    }
    // A>B && A=B && A<B

    if( A_Rank > C_Rank )
    {
        a = a--;
    }
    else if( A_Rank == C_Rank )
    {
        if( A_GD > C_GD )
        {
            a = a--;
        }
        else if( A_GD == C_GD )
        {
            if( A_goal > C_goal )
            {
                a = a--;
            }
            else if( A_goal < C_goal )
            {
                c = c--;
            }
        }
        else if( A_GD < C_GD)
        {
            c = c--;
        }
    }
    else if ( A_Rank < C_Rank )
    {
        c = c--;
    }
}

```

```

// A>C && A=C && A<C

if( A_Rank > D_Rank )
{
    a = a--;
}
else if( A_Rank == D_Rank )
{
    if( A_GD > D_GD )
    {
        a = a--;
    }
    else if( A_GD == D_GD )
    {
        if( A_goal > D_goal )
        {
            a = a--;
        }
        else if( A_goal < D_goal )
        {
            d = d--;
        }
    }
    else if( A_GD < D_GD )
    {
        d = d--;
    }
}
else if ( A_Rank < D_Rank )
{
    d = d--;
}
// A>D && A=D && A<D

```



```

if( B_Rank > C_Rank )
{
    b = b--;
}
else if( B_Rank == C_Rank )
{
    if( B_GD > C_GD )
    {
        b = b--;
    }
    else if( B_GD == C_GD )
    {
        if( B_goal > C_goal )
        {
            b = b--;
        }
        else if( B_goal < C_goal )
        {
            c = c--;
        }
    }
    else if( B_GD < C_GD )
    {
        c = c--;
    }
}
else if ( B_Rank < C_Rank )
{
    c = c--;
}
// B>C && B=C && B<C

```

```

if( B_Rank > D_Rank )
{
    b = b--;
}
else if( B_Rank == D_Rank )
{
    if( B_GD > D_GD )
    {
        b = b--;
    }
    else if( B_GD == D_GD )
    {
        if( B_goal > D_goal )
        {
            b = b--;
        }
        else if( B_goal < D_goal )
        {
            d = d--;
        }
    }
    else if( B_GD < D_GD )
    {
        d = d--;
    }
}
else if ( B_Rank < D_Rank )
{
    d = d--;
}
// B>D && B=D && B<D

```

```

if( C_Rank > D_Rank )
{
    c = c--;
}
else if( C_Rank == D_Rank )
{
    if( C_GD > D_GD )
    {
        c = c--;
    }
    else if( C_GD == D_GD )
    {
        if( C_goal > D_goal )
        {
            c = c--;
        }
        else if( C_goal < D_goal )
        {
            d = d--;
        }
    }
    else if( C_GD < D_GD )
    {
        d = d--;
    }
}
else if ( C_Rank < D_Rank )
{
    d = d--;
}
// C>D && C=D && C<D

```

```

// now, the value of rank store in a,b,c,d
// if a == 1 , team A is the NO.1
// judge each value in a,b,c,d then display their detailed information
if( a == 1 )
{
    cout << "1\tA" << "\tRANK " << A_Rank << "\tGOAL DIFFERENCE " << A_GD << "\tGOALS " << A_goal << endl;
}
if( b == 1 )
{
    cout << "1\tB" << "\tRANK " << B_Rank << "\tGOAL DIFFERENCE " << B_GD << "\tGOALS " << B_goal << endl;
}
if( c == 1 )
{
    cout << "1\tC" << "\tRANK " << C_Rank << "\tGOAL DIFFERENCE " << C_GD << "\tGOALS " << C_goal << endl;
}
if( d == 1 )
{
    cout << "1\tD" << "\tRANK " << D_Rank << "\tGOAL DIFFERENCE " << D_GD << "\tGOALS " << D_goal << endl;
}
// NO.1

if( a == 2 )
{
    cout << "2\tA" << "\tRANK " << A_Rank << "\tGOAL DIFFERENCE " << A_GD << "\tGOALS " << A_goal << endl;
}
if( b == 2 )
{
    cout << "2\tB" << "\tRANK " << B_Rank << "\tGOAL DIFFERENCE " << B_GD << "\tGOALS " << B_goal << endl;
}
if( c == 2 )
{
    cout << "2\tC" << "\tRANK " << C_Rank << "\tGOAL DIFFERENCE " << C_GD << "\tGOALS " << C_goal << endl;
}
if( d == 2 )
{
    cout << "2\tD" << "\tRANK " << D_Rank << "\tGOAL DIFFERENCE " << D_GD << "\tGOALS " << D_goal << endl;
}
// NO.2

```

```

if( a == 3 )
{
    cout << "3\tA" << "\tRANK " << A_Rank << "\tGOAL DIFFERENCE " << A_GD << "\tGOALS " << A_goal << endl;
}
if( b == 3 )
{
    cout << "3\tB" << "\tRANK " << B_Rank << "\tGOAL DIFFERENCE " << B_GD << "\tGOALS " << B_goal << endl;
}
if( c == 3 )
{
    cout << "3\tC" << "\tRANK " << C_Rank << "\tGOAL DIFFERENCE " << C_GD << "\tGOALS " << C_goal << endl;
}
if( d == 3 )
{
    cout << "3\tD" << "\tRANK " << D_Rank << "\tGOAL DIFFERENCE " << D_GD << "\tGOALS " << D_goal << endl;
}
// NO.3

if( a == 4 )
{
    cout << "4\tA" << "\tRANK " << A_Rank << "\tGOAL DIFFERENCE " << A_GD << "\tGOALS " << A_goal << endl;
}
if( b == 4 )
{
    cout << "4\tB" << "\tRANK " << B_Rank << "\tGOAL DIFFERENCE " << B_GD << "\tGOALS " << B_goal << endl;
}
if( c == 4 )
{
    cout << "4\tC" << "\tRANK " << C_Rank << "\tGOAL DIFFERENCE " << C_GD << "\tGOALS " << C_goal << endl;
}
if( d == 4 )
{
    cout << "4\tD" << "\tRANK " << D_Rank << "\tGOAL DIFFERENCE " << D_GD << "\tGOALS " << D_goal << endl;
}
// NO.4

return 0;
}

```

9. Testing:

```
football match result

A US B  2:1
A US C  1:4
A US D  2:2
B US C  3:1
B US D  4:2
C US D  1:1

1      B      RANK 6  GOAL DIFFERENCE 3      GOALS 8
2      C      RANK 4  GOAL DIFFERENCE 1      GOALS 6
3      A      RANK 4  GOAL DIFFERENCE -2     GOALS 5
4      D      RANK 2  GOAL DIFFERENCE -2     GOALS 5
Press any key to continue . . .
```