

Homework 2A

2021315385

이건 / Gun Daniel Lee

Problem explanation

- Calculate the scores of 20 soccer teams and sort in descending order
- Given inputs are:
 - A: points given for a Win
 - B: points given for a Tie
 - A 20 by 20 matrix where the results are represented as "W" (win), "T" (tie), "L" (lose), and "X" (disregarded)

Solution explanation

- Using a nested for loop, calculate the scores of the 20 teams
- After calculating each team's score, check whether the new team's score needs to be sorted (meaning it is greater than the previous team's score)
- If sorting is needed, use a while loop to sort before calculating the next team's score

Solution explanation

- A: points for a win
- B: points for a tie
- Results: The league result matrix (20 by 21 for the null character since input is received as a string)
- Scores: sorted array of the scores
- i, j: iteration variables

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b;
6      char results[20][21];
7      int scores[20] = {0, 0, };
8      int i, j;
9
```

Solution explanation

- Scan for A and B
- Use a for loop to scan for the results

```
9
10 scanf("%d %d", &a, &b);
11
12 for(i = 0; i < 20; i++)
13 {
14     scanf("%s", results[i]);
15 }
16
```

Solution explanation

- Nested for loop for calculating and sorting

```
17     for(i = 0; i < 20; i++)
18     {
19         for(j = 0; j < 20; j++)
20         {
21             if(results[i][j] == 'W')
22             {
23                 scores[i] += a;
24             }
25             else if(results[i][j] == 'T')
26             {
27                 scores[i] += b;
28             }
29         }
30         int index = i;
31         while(scores[index] > scores[index - 1] && index > 0)
32         {
33             int temp = scores[index];
34             scores[index] = scores[index - 1];
35             scores[index - 1] = temp;
36             index--;
37         }
38     }
```

Solution explanation

- Nested for loop for calculating and sorting
- Part 1: Calculating
 - For all the results in `results[i]`, add the score into `scores[i]`
 - Add *A* if it's a win and *B* if it's a tie

```
17     for(i = 0; i < 20; i++)
18     {
19         for(j = 0; j < 20; j++)
20         {
21             if(results[i][j] == 'W')
22             {
23                 scores[i] += a;
24             }
25             else if(results[i][j] == 'T')
26             {
27                 scores[i] += b;
28             }
29         }
```

Solution explanation

- Nested for loop for calculating and sorting

- Part 2: Sorting
 - Condition: current score is greater than previous score and current score is not the first score
 - Use while loop to keep swapping if condition is true

```
int index = i;
while(scores[index] > scores[index - 1] && index > 0)
{
    int temp = scores[index];
    scores[index] = scores[index - 1];
    scores[index - 1] = temp;
    index--;
}
```


Solution explanation

- Print results using for loop

```
40
41     for(i = 0; i < 20; i++)
42     {
43         printf("%d\n", scores[i]);
44     }
45 }
```

Solution analysis

- Main Idea was to reduce the sorting process by eliminating already sorted scores
- Favorable inputs:
 - The results are already sorted in descending order

Thank you!