COL100
Introduction to Computer Science
Assignment 4

1 Instructions

Dear Students, Assignment 4 has been released on Gradescope. Below, you'll find important details regarding the assignment, submission guidelines, and support channels for any queries.

1.1 Assignment Details:

Weightage: 8% of the overall grade
Submission Platform: Gradescope (https://www.gradescope.com/courses/561465)
Deadline: 4th October 2023, 11:55pm

1.2 Submission Guidelines:

Make sure to include comments in your code explaining your approach. When submitting, you need to upload files

1. `[kerberosID1]_[kerberosID2]_pragg_n_play.c`
2. `[kerberosID1]_[kerberosID2]_array_queries.c`

Please change the name of these files according to your group information. The starter code has been made available to you on https://replit.com/team/col100

The assignment must be submitted via **Gradescope** before the deadline. Late submissions will not be accepted. This assignment will be checked for plagiarism. Necessary action shall be taken against anyone found guilty of it.

1.2.1 Forming Groups

For assignment 4, you are allowed to work in groups of 2. This is optional. At the time of submission you will be given an option to link your submission with another student. You may do so by selecting their name from a dropdown. Additionally, all submitted documents should follow the given file naming scheme. **You should write both entry numbers as part of your submission c file.**

1.3 Support and Queries:

1. **All queries about this assignment should be posted only in the relevant Piazza board.** No other queries shall be entertained. Please read through the previously posted queries before posting.

2. **Before using any function from any existing library or using any algorithm not covered in the class, get a confirmation from the instructors by posting on Piazza.**

Additional Resources:

For a refresher on the concepts related to the assignment, you can refer to the course lecture notes, slides, and relevant sections from the textbook. Assignment Grading: Your assignments will be graded on correctness, efficiency of your code, proper usage of comments, and adherence to the submission guidelines. We wish you the best of luck with the assignment.

2 Pragg-n-Play

As you may be aware, Rameshbabu Praggnanandhaa, an Indian grandmaster recently got to the Chess World Finals where he faced Magnus Carlsen. Even though Praggnanandhaa lost the finals, he won the hearts of every Indian. Determined to excel, Pragg appraoched you to help him practice efficiently in chess.
You need to develop an engine for him to analyse the games better. Since Pragg is short on time, he has asked you to finish the project before **4th October 2023**.

2.1 Taking chess board as input and printing

Files to be changed: **utils.c, main.c**
Functions to be implemented: **printBoard(), main()**
For taking the board as input, you will be given 64 2-digt numbers. Each number corresponds to a square on the chess board. The first digit of the number represents the piece type. Following is the convention adapted:

```
EMPTY 0
PAWN 1
ROOK 2
KNIGHT 3
BISHOP 4
QUEEN 5
KING 6
```

The second digit of the number represents the color of the piece. Following is the convention for the colors:

```
EMPTY 0
WHITE 1
BLACK 2
```

So, if an entry is 61, that means the piece at that position is a White King.
Hint: For separating the type and color of the piece, you can use the technique from the "Sum of the Digits" problem from Lab-4.

You need to print the chess board as output. Following is the character convention for pieces:

```
EMPTY .
PAWN P
ROOK R
```

```
KNIGHT N
BISHOP B
QUEEN Q
KING K
ANY_OTHER_VALUE ?
```

To distinguish between black and white pieces, make it Uppercase for White pieces and lowercase for Black pieces.

2.1.1 Sample Input

22 32 42 52 62 42 32 22
12 12 12 12 12 12 12 12
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
11 11 11 11 11 11 11 11
21 31 41 51 61 41 31 21

2.1.2 Sample Output

r n b q k b n r
p p p p p p p p
. . . . . .
. . . . . .
. . . . . .
. . . . . .
P P P P P P P P
R N B Q K B N R

2.2 Evaluating the position

While analysing any chess problem, one of the main metric is the net material advantage for any player. Given a chessboard, you need to evaluate the position and return the winner and their net advantage. In case of 0 score, return "BALANCED" in place of the color. Following is the weight/importance of each piece:

```
PAWN 1
ROOK 5
KNIGHT 3
BISHOP 3
```

`QUEEN 9`

Note that we don't need the value for King as it will always be present on the board for both the players. For convenience, you can assign it to 0 for ease of implementation.

2.2.1 Sample Input

22 32 42 52 62 42 32 22
12 12 12 12 12 12 12 12
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
11 11 11 11 11 11 11 11
21 31 41 51 61 41 31 21

2.2.2 Sample Output

0 BALANCED
Here, it is the starting board. Hence, it is **balanced**.

2.2.3 Sample Input

22 32 42 52 62 42 32 22
12 12 12 12 12 12 12 12
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
11 11 11 11 11 11 11 11
21 31 41 51 61 41 31 21

2.2.4 Sample Output

8 BLACK
Here, at the bottom right, the knight and rook are missing for the white piece. Hence, the score is 5 + 3 for black.

3 Different Queries on Arrays

In this problem you will be given q number of queries , each query represent different array problem.

Query 1 : In this you will be given two integers n and k as inputs followed by n numbers of that array ,where n represent array size and k represent number of items you can chose from the given array.

You start with 0 score and you have to choose k items from array (of size n) each number/item you choose from array will be added to your score.You have to maximize your score.

Query 2 : In this you will be given 2 arrays and its sizes , both array are sorted but are in rotational state (i.e, the smallest element of each array might not start from 0th index in array).

You have to merge this two array into single array such that final array we obtained is also sorted.

Query 3 : In this you will be give a array of n numbers you have to create a 2-D binary matrix/array from these n numbers such that i column represent binary presentation of i index number in array where i $\in$ [0,n-1].

3.1 Constraints on Input:

Query 1 : $1 \leq n \leq 100, 0 \leq k \leq n, 0 \leq a_i \leq$1000 where a_i is ith index element of array.

Query 2 : $1 \leq n, m \leq 100, 0 \leq a_i \leq 1000$ and $0 \leq b_j \leq 1000$ where a_i and b_j are ith and jth index elements of first and second array respectively and i $\in$ [0,n-1] , j $\in$ [0,m-1]

Query 3 : $1 \leq n \leq 100, 0 \leq a_i \leq$1000 where a_i is ith index element of array.

3.2 Sample Input

3
1 5 3
10 0 3 10 5

2 5 3
5 6 1 2 3
4 8 9
3 5
3 4 5 1 8

3.3 Sample Output

25
1 2 3 4 5 6 8 9
0 0 0 0 1
0 1 1 0 0
1 0 0 0 0
1 0 1 1 0

3.4 Explanation

First line in Sample Input is **3** which represent number of queries.

Next line in Sample Input is **1 5 3** which represent **Query 1** where n=5 and k =3 .Its next line Sample Input has the array value ,let call that array **score =[10,0,3,10,5]**. Now for **Query 1** we have to maximize total score by selecting 3 items from array **score** , so we will select score[0],score[3],score[4] for maximum score. So maximum score obtain for **Query 1** will be score[0]+score[3]+score[4] = 10+10+5 = 25 . Hence Output for **1st Query** is **25**.

Next line in Sample Input is **2 5 3** which represent **Query 2** where n=5 and m =3. Next 2 lines of Sample Input contain array value of size 5 and 3 respectively say **a = [5,6,1,2,3] and b = [4,8,9]**. You can see that for both arrays **a** and **b** their elements are arranged in rotational ascending order for **a** you can start from index 2 and move till index 1 in circular manner then you will get **[1,2,3,5,6]** which is in ascending order. Similarly for **b** you can start from index 0 and move till index 2 you will get **[4,8,9]**.
So After merging this two rotational sorted array into single sorted array you will get **[1,2,3,4,5,6,8,9]**. Hence output for **Query 2** is **[1,2,3,4,5,6,8,9]**.

Next line in Sample Input is **3 5** which represent **Query 3** where n = 5. Next line contain n=5 numbers i.e., **[3,4,5,1,8]**. **Binary representation of 3,4,5,1 and 8 is 11,100,101,1 and 1000 respectively.** .

Now 2D array representation of these Binary numbers will be such that column1 =

0011 (1st column of 2D array, which represent 3) , column2 = 0100(2nd column of 2D array , which represent 4) ,column3 = 0101 (3rd column of 2D array , which represent 5), column4 =0001(4th column , which represent 1) and column5 = 1000 (5th column, which represent 8).

0 0 0 0 1
0 1 1 0 0
1 0 0 0 0
1 0 1 1 0

The Row number of 2D array/matrix will be decided by longest length of binary representation of given number. In our case we have $3 = (11)_2, 4 = (100)_2, 5 = (101)_2, 1 = (1)_2$ and $8 = (1000)_2$. As 8 has length of 4 (in binary representation) , hence number of rows in 2D array/matrix will be also 4.