CSCE 206 Spring 2020 Lab Assignment: #5

Submission Deadline: 23:59, April 27th, 2020 (Monday).

1. Follow the submission guidelines in the below link and make the submission through eCampus.
   https://prathiksha1995.github.io/CSCE206//Submission.html
2. Add comments to your code including your name, UIN and the class section you are in with block comments at the head of your code file.

Question 1. **Hex Calculator** (40 points)

Write a program to implement +, -, *, /, four operations between two hex numbers. The program is required to accept a single C string formula as input (10 points) and then properly separate two hex numbers and an operator from the input (10 points) and finally perform the operation according to the operator and return a correct decimal answer on terminal/console (10 points). Name your program file **Hw5_q1_code.c**.

Requirements are following:

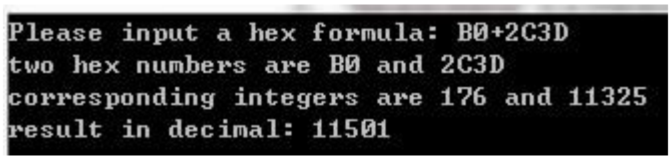
1. **Not allowed** to use any other libraries as your header files except <stdio.h> and <math.h>.

   Write the conversion function by yourself.

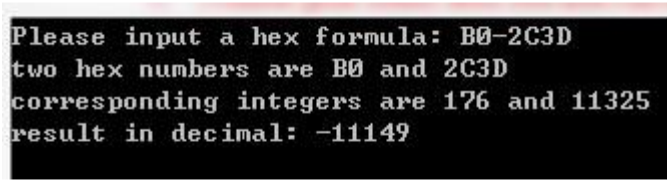
2. Use **scanf** function to accept input and input type must be stored as a character array variable.

3. You can pick some code from Lab 4.

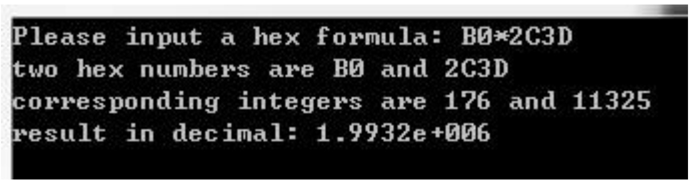
Sample inputs/outputs:



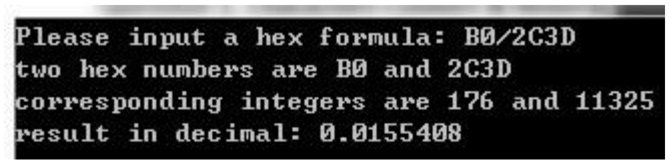
```
Please input a hex formula: B0+2C3D
two hex numbers are B0 and 2C3D
corresponding integers are 176 and 11325
result in decimal: 11501
```



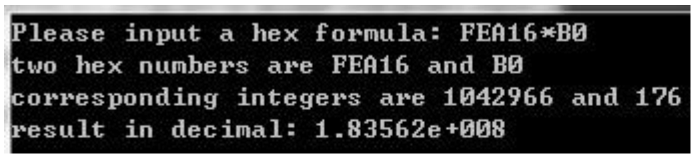
```
Please input a hex formula: B0-2C3D
two hex numbers are B0 and 2C3D
corresponding integers are 176 and 11325
result in decimal: -11149
```



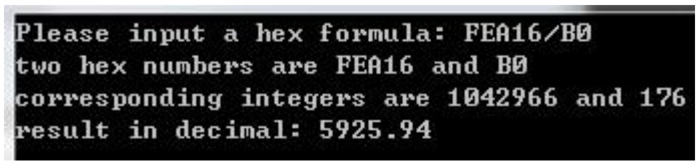
```
Please input a hex formula: B0*2C3D
two hex numbers are B0 and 2C3D
corresponding integers are 176 and 11325
result in decimal: 1.9932e+006
```



```
Please input a hex formula: B0/2C3D
two hex numbers are B0 and 2C3D
corresponding integers are 176 and 11325
result in decimal: 0.0155408
```



```
Please input a hex formula: FEA16*B0
two hex numbers are FEA16 and B0
corresponding integers are 1042966 and 176
result in decimal: 1.83562e+008
```



```
Please input a hex formula: FEA16/B0
two hex numbers are FEA16 and B0
corresponding integers are 1042966 and 176
result in decimal: 5925.94
```

Hint:

1. Go through class slides on pointers and arrays. Understand how a pointer works on an array and how to pass a C string to a function (call by reference or call by pointer);
2. Go through class slides on fundamental datatypes. Understand how to use switch statement and how to handle datatype conversions (integer to float/double and float/double to integer);
3. Borrow your code from Question 2, Lab4 and merge it as a hex2int or hex2double function so that it could be called inside main;
4. Properly separate those numbers and operator from the input C string.

Question 2. Counting Game (60 points)
Given a group of N players sitting in a circle on N chairs that have been labelled in order from $1 \sim N$. The player sitting on the k^{th} position $(k \leq N)$ starts the game by calling out the number 1 and the game goes in the clockwise direction until the number reaches M. Each time the player calling out the number M will leave the seat and the remaining players continue until one player is left. That is the person sitting on the k^{th} chair calls out 1, person sitting on the $(k+1)^{th}$ chair calls out 2, the person sitting on the $(k+2)^{th}$ chair calls out 3, so on and so forth, until the person sitting on the $(k+M-1)^{th}$ chair calls out M. Then the person sitting on the $(k+M-1)^{th}$ chair leaves the chair and the remaining $(N-1)$ players continue the game. The last player left will be the winner.
For example, if $N=5$, $k=2$, $M=3$, will have (* denotes a player),

Chair label	1	2	3	4	5
Before	*	*(saying 1)	*	*	*
1st round	*	*	*		*(saying 1)
2nd round	*		*(saying 1)		*
3rd round			*(saying 1)		*
4th round					*(winner)

if $N = 5$, $k = 5$, $M = 4$, will have (* denotes a player),

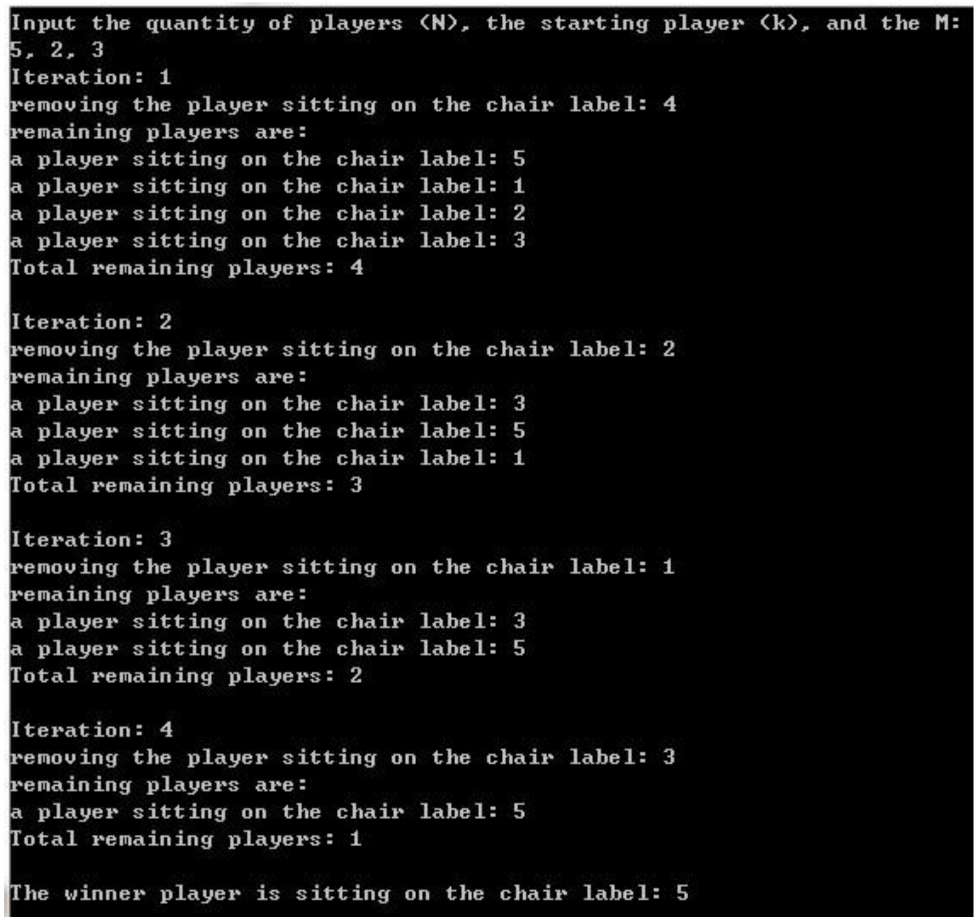
Chair label	1	2	3	4	5
Before	*	*	*	*	*(saying 1)
1st round	*	*		*(saying 1)	*
2nd round	*			*(saying 1)	*
3rd round	*				*(saying 1)
4th round					*(Winner)

Write a program, to accept three numbers as input for N, k, M, respectively and play this game and find the winner. Also print the real-time seating of the remaining players in each round (30 points). Name your program file **Hw5_q2_code.c**.

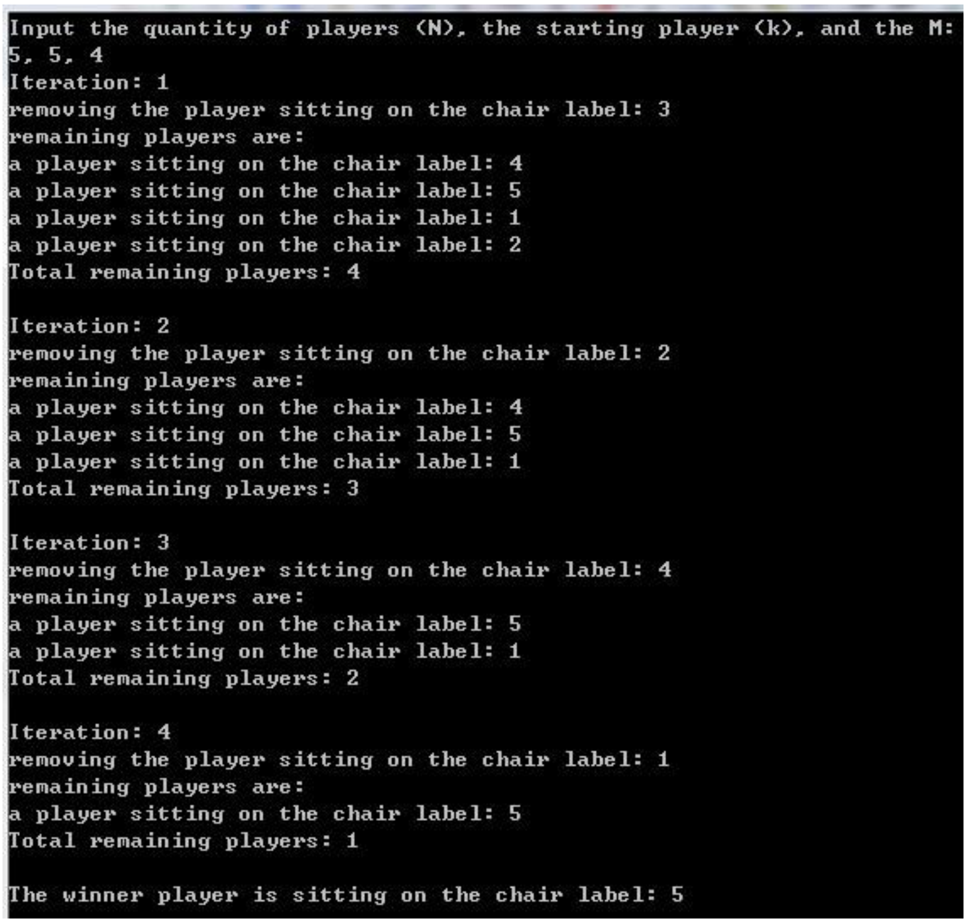
Requirements are following:

1. **Not allowed** to use any other libraries as the header files except <stdio.h> or <math.h> or <malloc.h>.

Sample inputs/outputs:



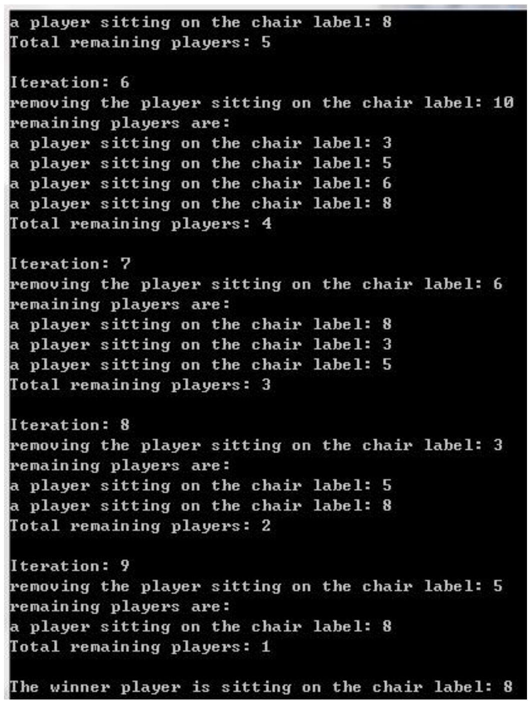
```
Input the quantity of players (N), the starting player (k), and the M:
5, 2, 3
Iteration: 1
removing the player sitting on the chair label: 4
remaining players are:
a player sitting on the chair label: 5
a player sitting on the chair label: 1
a player sitting on the chair label: 2
a player sitting on the chair label: 3
Total remaining players: 4

Iteration: 2
removing the player sitting on the chair label: 2
remaining players are:
a player sitting on the chair label: 3
a player sitting on the chair label: 5
a player sitting on the chair label: 1
Total remaining players: 3

Iteration: 3
removing the player sitting on the chair label: 1
remaining players are:
a player sitting on the chair label: 3
a player sitting on the chair label: 5
Total remaining players: 2

Iteration: 4
removing the player sitting on the chair label: 3
remaining players are:
a player sitting on the chair label: 5
Total remaining players: 1

The winner player is sitting on the chair label: 5
```



```
Input the quantity of players (N), the starting player (k), and the M:
5, 5, 4
Iteration: 1
removing the player sitting on the chair label: 3
remaining players are:
a player sitting on the chair label: 4
a player sitting on the chair label: 5
a player sitting on the chair label: 1
a player sitting on the chair label: 2
Total remaining players: 4

Iteration: 2
removing the player sitting on the chair label: 2
remaining players are:
a player sitting on the chair label: 4
a player sitting on the chair label: 5
a player sitting on the chair label: 1
Total remaining players: 3

Iteration: 3
removing the player sitting on the chair label: 4
remaining players are:
a player sitting on the chair label: 5
a player sitting on the chair label: 1
Total remaining players: 2

Iteration: 4
removing the player sitting on the chair label: 1
remaining players are:
a player sitting on the chair label: 5
Total remaining players: 1

The winner player is sitting on the chair label: 5
```

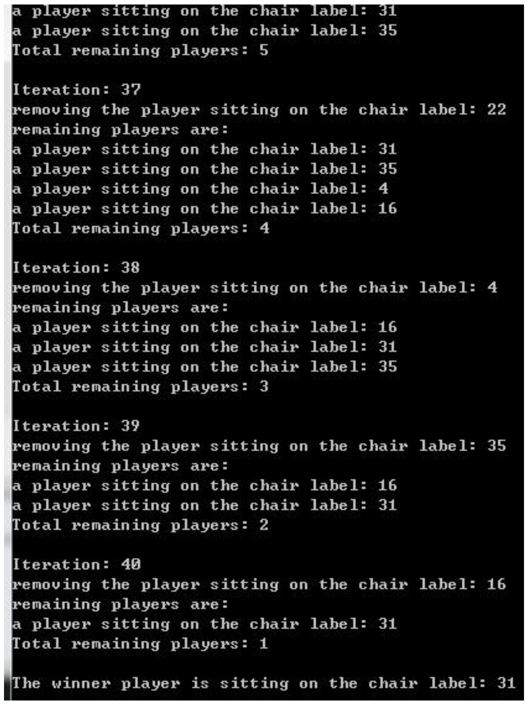
More test cases:

$N = 10, k = 2, M = 11$ => Winner: 8



```
a player sitting on the chair label: 8
Total remaining players: 5

Iteration: 6
removing the player sitting on the chair label: 10
remaining players are:
a player sitting on the chair label: 3
a player sitting on the chair label: 5
a player sitting on the chair label: 6
a player sitting on the chair label: 8
Total remaining players: 4

Iteration: 7
removing the player sitting on the chair label: 6
remaining players are:
a player sitting on the chair label: 8
a player sitting on the chair label: 3
a player sitting on the chair label: 5
Total remaining players: 3

Iteration: 8
removing the player sitting on the chair label: 3
remaining players are:
a player sitting on the chair label: 5
a player sitting on the chair label: 8
Total remaining players: 2

Iteration: 9
removing the player sitting on the chair label: 5
remaining players are:
a player sitting on the chair label: 8
Total remaining players: 1

The winner player is sitting on the chair label: 8
```

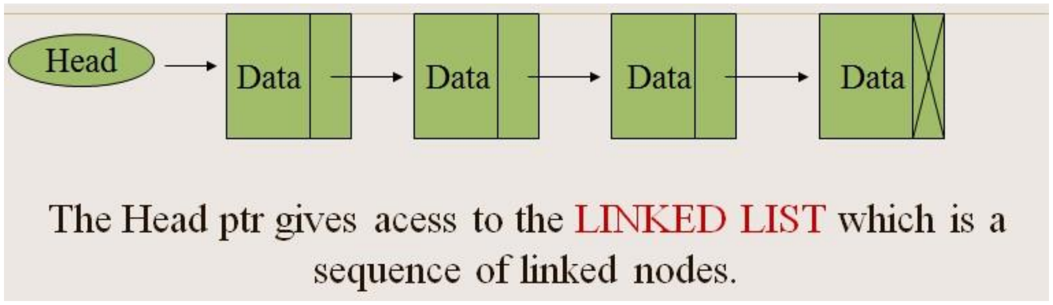
$N = 41, k = 1, M = 3$ => Winner: 31



```
a player sitting on the chair label: 31
a player sitting on the chair label: 35
Total remaining players: 5

Iteration: 37
removing the player sitting on the chair label: 22
remaining players are:
a player sitting on the chair label: 31
a player sitting on the chair label: 35
a player sitting on the chair label: 4
a player sitting on the chair label: 16
Total remaining players: 4

Iteration: 38
removing the player sitting on the chair label: 4
remaining players are:
a player sitting on the chair label: 16
a player sitting on the chair label: 31
a player sitting on the chair label: 35
Total remaining players: 3

Iteration: 39
removing the player sitting on the chair label: 35
remaining players are:
a player sitting on the chair label: 16
a player sitting on the chair label: 31
Total remaining players: 2

Iteration: 40
removing the player sitting on the chair label: 16
remaining players are:
a player sitting on the chair label: 31
Total remaining players: 1

The winner player is sitting on the chair label: 31
```

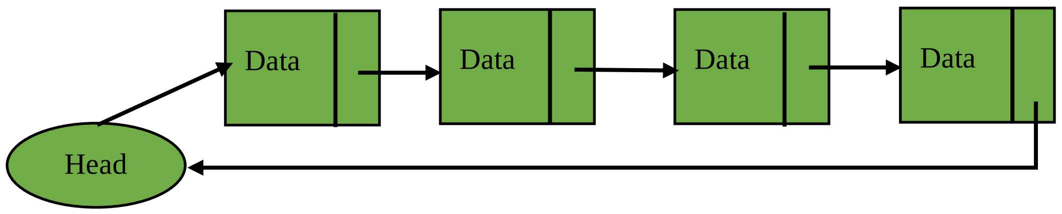
Hint:

1. There are two ways to go: 1) Using a single-dimension array;
   2) Using the struct type in C.
   Basically, (1) will refer to Ch9 slide talking about Array and (2) will refer to Ch12 slide talking about C structure. However, both will probably refer to Ch8 slide talking about Pointer.
   If you choose (1), you might want to consider two things: a. How to denote a player in an array who leaves his or her seat? b. How to loop on an array (or in other words, which operator in C could result in a loop? Check the operators slide).
   If choose (2), you might want to use a circular linked list (linked list containing a cycle, check below, it is a linked list with its tail to the head) and primitive list functions (insert a node, delete a node). Be careful when you delete a node. You will have to use *free(node)* in <malloc.h> so that the memory could be properly recycled. Tutorial is here.
2. Below is a linked list captured from the C structure slide.



The Head ptr gives acess to the LINKED LIST which is a sequence of linked nodes.

The only difference between the linked list cycle and the linked list is the linked list cycle will have its tail and head connected while the linked list has its nulled tail only.


Data
Data
Data
Data
Head