```
//Name: Mehmet Fatih Çelik
//ID: 2385268

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define r 5
#define c 4

void displayBoard(int [r][c]);
int randomPowerTwo(int);
int shootMerge(int [r][c], int);
int fullorNot(int [r][c]);

int main(){
        srand(time(NULL));
        printf("*Shoot Merge*\n");
        printf("Lets get started!\n");
        int board[r][c];
        int randNum = randomPowerTwo(32), maxValue, numShoots=0, i, j;

        for(i=0;i<r;i++){ // make all zero
                for(j=0;j<c;j++){
                        board[i][j] = 0;
                }
        }

        char myChoice = 'Y';

        displayBoard(board);
```

```
while(myChoice == 'Y'){ // game started

    maxValue = shootMerge(board, randNum);
    randNum = randomPowerTwo(maxValue);

    displayBoard(board);
    numShoots++;
    printf("%d shot so far!\n\n", numShoots);

    if (fullorNot(board)){ // if board full already
        printf("The board is full now with %d shots!\n", numShoots);
        printf("Game over!\n");
        printf("Would you like to play again (Y/N)? ");
        fflush(stdin);
        scanf("%c",&myChoice);
        if (myChoice == 'N'){
            printf("Byeee!\n");
            return 0;
        }
        else if(myChoice == 'Y'){
            printf("\n\n");
            for(i=0;i<r;i++){ //make all zero
                for(j=0;j<c;j++){
                    board[i][j] = 0;
                }
            }
            numShoots = 0;

            maxValue = shootMerge(board, randNum);
            randNum = randomPowerTwo(maxValue);
            displayBoard(board);
            numShoots++;
            printf("%d shot so far!\n\n", numShoots);
        }
        else{
            int controller = 1;

            while(controller){

                printf("\nThat is not valid, please try again!\n\n");
                printf("Would you like to play again (Y/N)? ");
                fflush(stdin);
                scanf("%c",&myChoice);
                if (myChoice == 'Y'){
                    printf("Lets get started!\n");
                    controller = 0;

                    for(i=0;i<r;i++){ //make all zero
                        for(j=0;j<c;j++){
                            board[i][j] = 0;
                        }
                    }
                    numShoots = 0;

                    randNum = randomPowerTwo(rand()%33);
                    maxValue = shootMerge(board, randNum);
                    displayBoard(board);
                    numShoots++;
                    printf("%d shot so far!\n\n", numShoots);
                }
```

```
                    if (myChoice == 'N'){
                        printf("Byeee!\n");
                        return 0;
                    }

                }
            }
        }
    }
    return 0;
}

void displayBoard(int board[r][c]){ // I didnt send input rows, and columns ,as I used define
    int i,j;
    printf("\tC1\tC2\tC3\tC4\n");

    for (i = 0; i < 5; i++) {
        printf("\t");
  for (j = 0; j < 4; j++){
    if (board[i][j] != 0)
                printf("%d\t", board[i][j]);
            else
                printf("\t");
    }
    printf("\n");
 }
}

int randomPowerTwo(int maxValue){
    int counter = 0;
    while(maxValue != 0){ // how many 2 it has
```

```
                maxValue /= 2;
                counter++;
        }

        return pow(2,(rand()%(counter)));
}
int shootMerge(int board[r][c], int randNum){
        int choice, row, controller = -99, i ,j, maxValue = 0;
        printf("You have %d, which column you like to shoot [1-4]? ",randNum);
        scanf("%d",&choice);

        if (choice>0 && choice <5){
                choice--; // indexs starts from 0
                for(i=4;i>=0;i--){ // starts from 4 because there is 5 rows, last one's index 4, I am controlling if it is full or not.
                        if (board[i][choice] != 0) // if it is not empty, make controller = 0 and break.
                                break;
                        controller = i;
                }

          if (controller == -99) { // if controller is comes out -99(is not changed) here, column is full.
                        printf("Column is full, try again!\n");
       return shootMerge(board, randNum);
                }

                board[controller][choice] = randNum; // if column is empty to shoot, we shot.

                for(;controller-1 >= 0;controller--){
                        if (board[controller][choice] == board[controller - 1][choice]) { // if we shoot a column, and it has the same value in the prev item, we need to merge.
                                board[controller - 1][choice] += board[controller][choice]; // merging
         board[controller][choice] = 0; // we merged, last index is 0
```

```
                        }
                        else
                                break; // if there is no need to be done merging, exit the loop.
                }
        }

        else{
                printf("Sorry, that is not a valid column, try again!\n");
                return shootMerge(board, randNum);
        }

        for (i = 0; i < 5; i++) { // finding maxValue for returning
    for (j = 0; j < 4; j++) {
      if (board[i][j] > maxValue)
        maxValue = board[i][j];
    }
  }

        return maxValue;
}

int fullorNot(int board[r][c]){
        int controller = 1, i, j;

        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        if (!board[i][j]) // if empty
                                controller = 0;
                }
        }
        return controller;
}
```