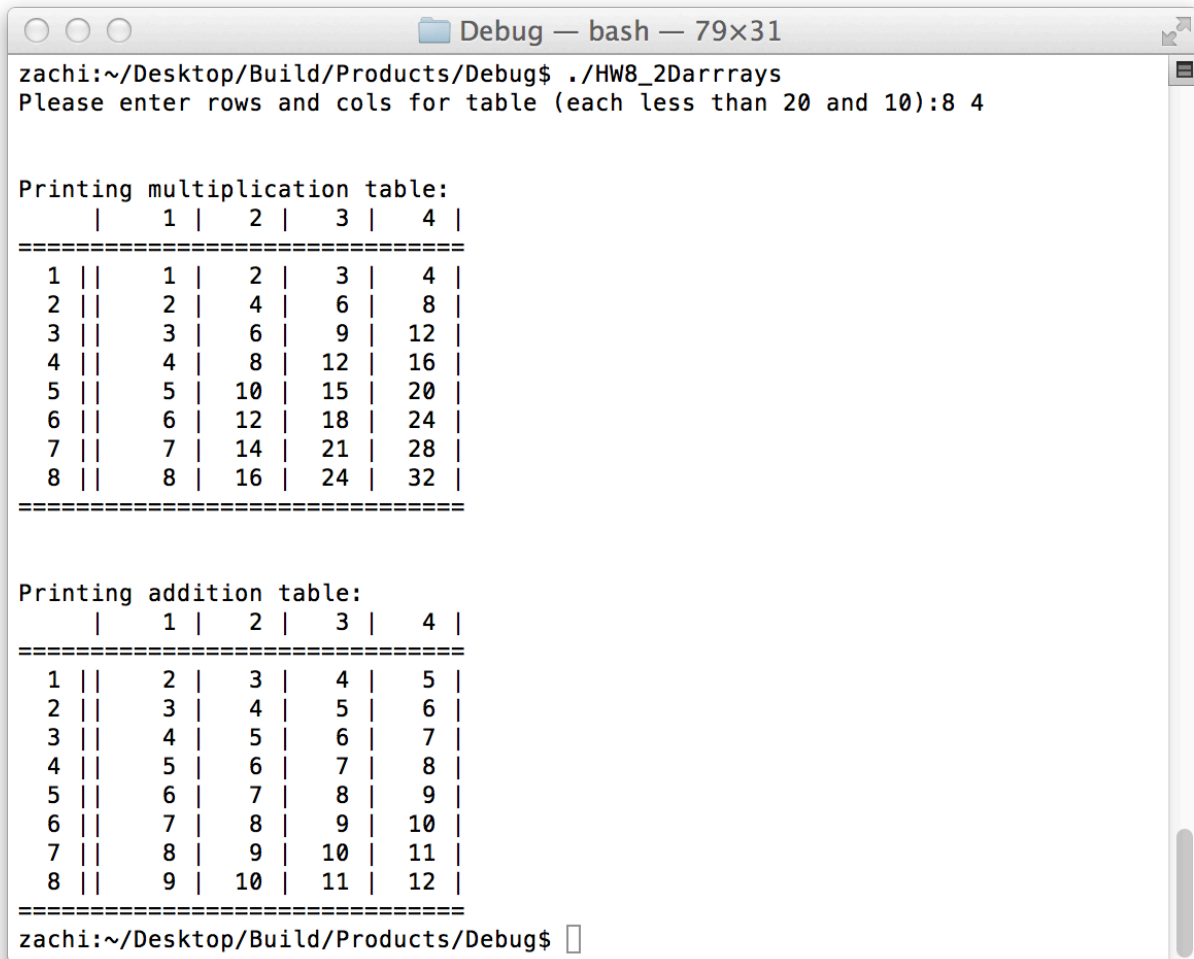


Assignment HW9
Due date (on or before): Announced in class.

1. 2D arrays.

Write a program that prompts the user for two integers, Rows and Cols, and prints out a nice multiplication table and addition table with the specified number of rows and cols. Use three functions as described in the code skeleton below.

See example screen shots below, and also the skeletal code.



```
zachi:~/Desktop/Build/Products/Debug$ ./HW8_2Darrays
Please enter rows and cols for table (each less than 20 and 10):8 4

Printing multiplication table:
=====
| 1 | 2 | 3 | 4 |
=====
1 || 1 | 2 | 3 | 4 |
2 || 2 | 4 | 6 | 8 |
3 || 3 | 6 | 9 | 12 |
4 || 4 | 8 | 12 | 16 |
5 || 5 | 10 | 15 | 20 |
6 || 6 | 12 | 18 | 24 |
7 || 7 | 14 | 21 | 28 |
8 || 8 | 16 | 24 | 32 |
=====

Printing addition table:
=====
| 1 | 2 | 3 | 4 |
=====
1 || 2 | 3 | 4 | 5 |
2 || 3 | 4 | 5 | 6 |
3 || 4 | 5 | 6 | 7 |
4 || 5 | 6 | 7 | 8 |
5 || 6 | 7 | 8 | 9 |
6 || 7 | 8 | 9 | 10 |
7 || 8 | 9 | 10 | 11 |
8 || 9 | 10 | 11 | 12 |
=====
zachi:~/Desktop/Build/Products/Debug$
```

```
Debug — bash — 79x25
zachi:~/Desktop/Build/Products/Debug$ ./HW8_2Darrays
Please enter rows and cols for table (each less than 20 and 10):5 10

Printing multiplication table:
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
=====
1 || 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
2 || 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
3 || 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
4 || 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
5 || 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
=====

Printing addition table:
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
=====
1 || 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
2 || 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
3 || 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
4 || 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
5 || 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
=====
zachi:~/Desktop/Build/Products/Debug$
```

Code skeleton:

```
int main(void)
{
    int a[ROWS][COLS];
    int rows, cols;

    printf("Please enter rows and cols for table (each less than
%d and %d):", ROWS, COLS);
    scanf("%d%d",&rows,&cols);

    // Multiplication table
    printf("\n\nPrinting multiplication table:\n");
    multTable(a, rows, cols); // Fills a[][]
    printArray(a, rows, cols);

    // Addition table
    printf("\n\nPrinting addition table:\n");
    addTable(a, rows, cols); // Fills a[][]
    printArray(a, rows, cols);
}
```

```
}    return 0;
```

```
***** Start of code + Screen shot*****
```

```
***** End of code + Screen shot*****
```

2. Black Jack.

In class we wrote a skeleton program to play BlackJack. In the following two exercises you'll improve on the program we wrote. Use that program as a base, and modify it as needed.

Here are some guidelines. Then, a few screen shots from the result, AND then a description of the function main() I used. This will give you a hint as to which functions to use to organize your program, what parameters they take, and so on.

So these are the suggested guidelines to get from what we did in class to this much cleaner program.

- a. Create a function printHands(...)
- b. Move things to 'global' scope.
- c. Create function playRound()
- d. Create a function play0Round() that plays the first time (getting each player two cards)
- e. Build a REAL deck, with 52 cards, so there's no option of getting 5 of the same card in one game!!
- f. Build a function anyBusted()
- g. Build functions strategy1() and strategy2(), each returns an integer, and these represent the strategy of each player.

Again, to get a feeling of how your main() should look like, see below.

Some screenshots:

Just a regular game:

```
zachi:~/Desktop/Build/Products/Debug$ ./HW9_Casino22
Welcome to playing (almost) BlackJack game.
You are getting 2 cards initially, and your goal is to get their sum to 21.
Each picture card is worth 10 points. Ace is counted as 1.

**** Printing hands:

Player 1 cards:  (value=6)
3 3
Player 2 cards:  (value=11)
6 5
*****

Do you want to take another card? [y or n]y
House is taking one card

**** Printing hands:

Player 1 cards:  (value=16)
3 3 10
Player 2 cards:  (value=12)
6 5 A
*****

Do you want to take another card? [y or n]y
House is taking one card

*****Final Status:

**** Printing hands:

Player 1 cards:  (value=19)
3 3 10 3
Player 2 cards:  (value=21)
6 5 A 9
*****

Player2 won.
zachi:~/Desktop/Build/Products/Debug$
```

Player 1 is busted. Player2 doesn't need to take a card.

```
Debug — bash — 79x28
zachi:~/Desktop/Build/Products/Debug$ ./HW9_Casino22
Welcome to playing (almost) BlackJack game.
You are getting 2 cards initially, and your goal is to get their sum to 21.
Each picture card is worth 10 points. Ace is coutned as 1.

**** Printing hands:

Player 1 cards:  (value=13)
8 5
Player 2 cards:  (value=13)
3 Q
*****

Do you want to take another card? [y or n]y

*****Final Status:

**** Printing hands:

Player 1 cards:  (value=-1)
8 5 9
Player 2 cards:  (value=13)
3 Q
*****

Player2 won.
zachi:~/Desktop/Build/Products/Debug$
```

```
// main
int main(void)
{

    int change = 0;

    int val1, val2;

    printf("Welcome to playing (almost) BlackJack game.\n");
    printf("You are getting 2 cards initially, and your goal is
to get their sum to 21.\n");
    printf("Each picture card is worth 10 points. Ace is coutned
as 1.\n");

    srand((unsigned int) time(NULL));

    play0Round(); // this includes shuffling/resetting the
deck
```

```

    printHands();

    // no hand can be busted after FIRST round

    change = playRound(); // change is '1' if any of the
    players took cards.

    if (change && !anyBusted())
    {

        printHands();

        change = playRound();

    }

    // we allow two rounds maximum

    printf("\n\n*****Final Status:\n");
    printHands();

    // determine who won
    val1 = evalHand(p1,l1);
    val2 = evalHand(p2,l2);
    //printf("\nval1=%d, val2=%d\n.",val1,val2);

    if (val1>val2) printf("\nPlayer1 won.\n");
    if (val1<val2) printf("\nPlayer2 won.\n");
    if (val1==val2) printf("\nDraw.\n");

    return (0);
}

```

***** Start of code + Screen shot*****

***** End of code + Screen shot*****

3. Black Jack. (simulation)

Now that the game mechanics works, let's do some simulations!!

'Move' your main() into a 'playGame()' function that merely returns the result of the game.

Then, in main(), create a loop to call this function N times, and then average the result. See code example (and screen shot) below.

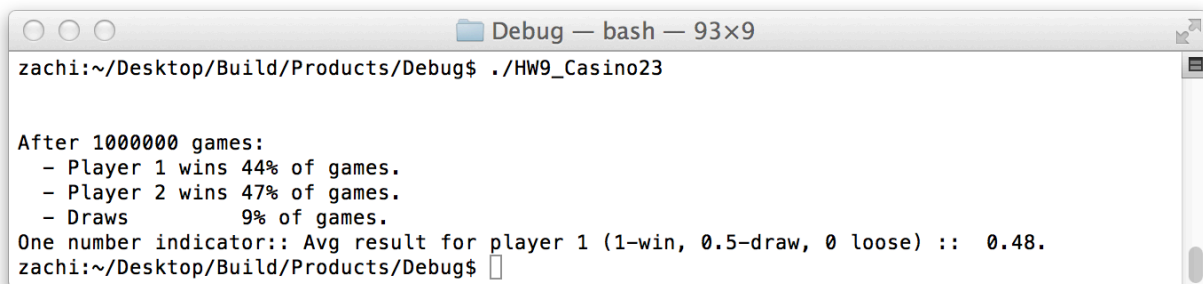
ONE important thing you will need to do: REMOVE the printouts and questions for the user!!

For example, in strategy1() you probably asked the user if, and how many, cards to take. Now, for simulation, you cannot allow that. The strategy has to be fully programmed in!

Run the program a few times to get a feeling for the best strategy, chances, and so on.

A few questions:

1. If both players have the same strategy, the second player (house) is still doing better. Why?
2. What is the best strategy for player1? What percentage did you get?
3. Are you sure this is the best strategy? Any thoughts on how to improve on that?



```
zachi:~/Desktop/Build/Products/Debug$ ./HW9_Casino23

After 1000000 games:
- Player 1 wins 44% of games.
- Player 2 wins 47% of games.
- Draws          9% of games.
One number indicator:: Avg result for player 1 (1-win, 0.5-draw, 0 loose) :: 0.48.
zachi:~/Desktop/Build/Products/Debug$
```

Below is an example of how the two basic functions, main() and playGame(), should look like. Please do NOT modify main() at all. As for playGame(): It is a good idea to leave it as is, but if you feel a real urge to change anything there, be my guest.

```
// Function playGame() returns:
//     1 - player1 won,
//     2 - player2 won,
//     0 - draw.
int playGame(void)
{

    int change = 0;
    int val1, val2;
    int res=0;
    static int firstTime = 1;

    if (firstTime)
    {
```



```

    firstTime = 0;
    srand((unsigned int) time(NULL));
    //srand((unsigned int) 0); // for Debugging only
}
resetDeck();

play0Round();

change = playRound();

if (change && !anyBusted())
    change = playRound();

val1 = evalHand(p1,l1);
val2 = evalHand(p2,l2);

// --> printHands(); // for Debugging only

if (val1>val2) res = 1;
if (val1<val2) res = 2;
if (val1==val2) res = 0;

return (res);
}

// main()
int main(void)
{
    int N = 1e6; // iterations
    int ii, val;
    int draw=0, win1=0, win2=0;

    for (ii=0; ii<N; ++ii) {
        val = playGame();
        switch (val) {
            case 0:
                draw++; break;
            case 1:
                win1++; break;
            case 2:
                win2++; break;
            default:
                break;
        }
    }
}

```

```

printf("\n\n");
printf("After %d games:\n", N);
printf(" - Player 1 wins %d%% of games.\n", ((win1*100)/N));
printf(" - Player 2 wins %d%% of games.\n", ((win2*100)/N));
printf(" - Draws          %d%% of games.\n", (100 -
(win1*100)/N - (win2*100)/N));
printf("One number indicator:: Avg result for player 1 (1-
win, 0.5-draw, 0 loose) ::
%5.2f.\n", (draw*0.5+win1)/(float)(draw+win1+win2) );
}

```

***** Start of code + Screen shot*****

***** End of code + Screen shot*****