TATYANA VLASKIN
WEEK 6 REPORT

Problem 1:

UNDERSTANDING:

1. We need to write a function that create 1D array of int values
2. Function should ask the user to input values for the array
3. If the user makes invalid entry (character, word, space floating point number), the user is notified that the inputs is invalid and is asked for made a differently entry
4. Once the user finishes entering the number, the function computes sum of the entered values
5. We are allowed to decide how large the array is.

DESING:

1. I will ask the user to enter 5 values. So I'll globally declare/define variable: const int SIZE = 5 that will represent size of the array.
2. I am thinking about writing functions: one for populating the array and another one for computing sum.
   a. void populateArray(double array[], int SIZE);
   b. double sumArray(const double array[], int SIZE)
3. When the user enters number, the array will be changed with every single entry; however, one all the numbers are entered, we do not want to use accidently change the array, so Ill add "const" in front of the array argument.
4. I am defining array as the double type because the user will be allowed to enter a double, but in case double is entered the user will get an invalid entry message. I am planning to check for valid entry as : while (cin.fail() || (array[i] != static_cast<int>(array[i]))). The plan is to take an entry made by the user, cast the entry into the int type and compare the original entry to the casted, if they are not equivalent, flag it as inva;id entry.
5. The void populateArray(double array[], int SIZE) will be a for loop:
   ```
    for (int i = 0; i < SIZE; i++)
   {    cin >> array[i]; }
   ```
6. The double sumArray(const double array[], int SIZE) function will be another for loop: entered
   ```
   int sum = 0;
   ```

```
for (int i = 0; i < SIZE; i++){         sum += array[i];       }
 return sum;
}
```

7. The double sumArray function will return sum

Problem 2:
UNDERSTANDING
We need to write a function that accepts two references to c-style strings and returns whether they have the same values. My understanding of this is problem is that we need to:

1. declare 2 variables of c-type string which is an array.
2. ask the user to enters 2 strings
3. compare 2 c-type strings (arrays) using strcmp(string1, string2) function

The comparison is true if the strings do not match. The function strcmp compares characters in C-string arguments a character at a time. If at any point the numeric encoding of the character from String1 is less than a numeric encoding of the corresponding character from String2 , the testing stops and a negative number is returned. If its other way around, a positive number is returned. If strings are the same, zero is returned. The logic is opposite of the bool (The comparison is true if the strings do not match.) I think this is a potential test question.

DESIGN:

1. Decide what will be the size of the array. I hope that none of the user will enter a string that is longer than 100 characters, so I'll chose 100 for size
2. Globally define size variable. const int SIZE = 100; For some reason in the book they keep putting const in front of the size, so I'll do the same thing, but honestly I do not think that I understand this.
3. Define 2 arrays/C-strings: char string1[SIZE] and   char string2[SIZE]
4. I think similar to the previous problem, I'll make 2 functions. One for the user to input 2 strings and the second one to compare two strings. No restriction on the input, user can enter whatever they want.
5. First function will be void userInput(char (&string1)[SIZE], char (&string2)[SIZE]). When you use C-string as a parameter in a function, you need to indicate size. I got this idea from page 376 in the book : int main(int

argc, char *argv[]) . Nothing special about this function. Simple cout and cin.getline. Ill use cin.getline in case string will have more than 1 word.

6. Second function will be void stringCompare (char (&string1)[SIZE], char (&string2)[SIZE])
7. I’ll define another variable: int comparison = strcmp(string1, string2)
8. Depending on the value of the int comparison there are will be if/else if/else statement:

```
if (comparison > 0){ string 1 is greater than string2}
else if (comparison == 0  ) { strings are equal}
else{ string1 is  less than string2}
```

Problem 3:

UNDERSTANDING

We need to write a program that does the following:

1. Allows the user to enter grades for each student.
2. As the grades are being entered, the program will count, using an array each grade. Possible grades are (0,1,2,3,4,5).
3. There is a hint that this can be done, by making an array of size 6, where each array element is initialized to zero. Whenever a zero is entered, increment the value in the array at index 0.
4. Whenever a one is entered, increment the value in the array at index 1, and so on, up to index 5 of the array.
5. The program should be capable of handling an arbitrary number of students. I think I’ll make a function where I’ll ask a user how many students took the quiz.

   DESING:

   1. Ask the user how many students took the quiz (variable students)
   2. Declare an array for 6 grades (0-5)
   3. Initialize each index in the array to zero
   4. Make a function to check that the input is valid (integer)
   5. For loop to enter grade for each student and count each grade
   6. For counting a grade, Ill make a function. In the description of the problem there is a hint: Whenever a zero is entered, increment the value in the array at index 0. Whenever a one is entered, increment the value in the array at index 1, and so on, up to index 5 of the array. I am thinking about making 6 if/else if statements and put them in the function:

      ```
      if (grade == 0)
      ```

```
   grades[0]++;
else if (grade == 1)
   grades[1]++;
else if (grade == 2)
   grades[2]++;
else if (grade == 3)
   grades[3]++;
else if (grade == 4)
   grades[4]++;
else if (grade == 5)
   grades[5]++;
```

7. For loop to print histogram

Problem 4:
UNDERSTANDING
We need to write a program that allows the user to play tic-tac-toe using 1D array.

1. The problem with ask the user for the names of the players
2. Once the names are entered the program will ask each player to enter their move, by specifying position on the board. The layout of the board is:
   ```
   1 2 3
   4 5 6
   7 8 9
   ```
3. After each move, the program displays the changed board with the information which positions are already filled, maked as X or O
4. Is the user makes a move that already taken, there should be a message letting the user know that that position is taken
5. If the user makes invalid entry, such as enters a position that does not exist, there should be an invalid error message
6. After move 5, there is a chance that there is a winner, so we need to check for winning combinations for each user and if such exists display a winning message
7. If full board is filled and there is no winner, a message is displayed that there is not winner
   DESIGN;

1. Ask for 2 names
2. Declare array: char cells[9] = {'-', '-', '-', '-', '-', '-', '-', '-', '-'}
3. Ask each player to make their entry. 1st user will be X and second user will be O. Thus X will have odd moves and O will have even moves. So I can do something like this:

```
for (int move = 1; move <= 9; move++) {
         if (move % 2) {
        // player 1 will start first, so he/she will have odd moves, so
        move % 2 = 1 for player 1
        Ask the user for the move
        cin >> cell
        }
        else {// if remainder is not 1
        Ask the second user for the move
        cin >> cell}
```

This will keep track of the moves, who made which entry.

4. After 5th move , there is potentially a winner, so I need to check for a winner. I need to check horizontal, vertical and diagonal positions.
5. To check for the winner, I need to check the following indexes/cells in the array:

```
//horizontal combinations//0-1-2, 3-4-5 и 6-7-8
//vertical combination//0-3-6, 1-4-7 и 2-5-8
//diagonal combination // 0-4-8 and 2-4-6
```

   If one of the above mentioned combinations has the same character, this indicates that there is a winner. I need to retrieve character of the winner

6. In the subsection 3, I indicated that move % 2 = 1 for X, so I'll introduce another variable and set it equal to 1.

```
if (number == 1){//fist player uses X, second one uses O
      cells[cell-1] = 'X';
   }
  else {
      cells[cell-1] = 'O';
   }
```

   This together with the following algorithm will be used to determine the winner:

```
  for(int i = 0; i < 3; i++) //3 cells need to be filled in for winning
```

```
    if ((cells[i*3] == cells[i*3+1]) && (cells[i*3+1] == cells[i*3+2]))
  //horizontal combinations//0-1-2, 3-4-5 and 6-7-8
  return cells[i];
  else if (cells[i] == cells[i+3] && cells[i+3] == cells[i+6])
   //vertical combination//0-3-6, 1-4-7 and 2-5-8
return cells[i];
  else if ((cells[0] == cells[4] && cells[4] == cells[8]) || (cells[2] ==
  cells[4] && cells[4] == cells[6]))
return cells[i]; //diagonal combination // 0-4-8 and 2-4-6
  else
  return '-'; //no winner
```

7. If there is no winner and move > 9, there is a message that game is over and there that nobody won.

Problem 5

UNDERSTANDING:

We need to write a program that allows the user to play tic-tac-toe using 2D array.

This is identical problem is identical to problem 4. The only difference is that we need to use 2d arrays. Look at the understanding section in problem 4.

DESIGN:

Declare a [3][3] array.

Ask the user to enter moves. After move 5 start checking to winner using the following algoriphm:

```
bool win()
{
   for (int i(0); i < 3; i++)
   {
      if ((table[i][0] == table[i][1]) && (table[i][0] == table[i][2]))
         return true;
      else if ((table[0][i] == table[1][i]) && (table[0][i] == table[2][i]))
         return true;
      else if ((table[0][0] == table[1][1] && table[0][0] == table[2][2]) ||
(table[2][0] == table[1][1] && table[2][0] == table[0][2]))
         return true;
   }
   return false; // if there is no winning combination
```

}

See comments, for more information.

I got a lot of help of this problem

I went to UW study center and one of the TAs spent over 30 min explaining to me how to do this problem. After that I spent over 10 hours trying to figure out this problem and make sense of all his recommendations.

PROJECT:

UNDERSTANDING:

1. Make array of words
2. Randomly pick number that will indicate how many words to select from the array of words
3. Randomly pick words from the array and put them in the string
4. Once the word is valid and stored as a variable:
5. User is notified how many guesses they have
6. String that needs to be guessed is presented to the user as '-' . Each '-' represents a letter in the word
7. List the valid letters to help the user (letter in the alphabet)
8. User is asked to guess a string one letter at a time
9. If the user guesses a letter, program shows the user what of the phrase has been correctly guessed so far. If the letter was guessed correctly, '-' character will be replaced with the correct letter. At the same time, list of the valid letters will be updated
10. If the user makes incorrect guess, the user is notified that they made incorrect guess and number of misses is counted, guess is decremented. List of the valid letters will be updated
11. If the user makes a guess that is correct but is the repeated guess of the same character, the user is notified that the letter was already guessed. I think in this case I will not consider this as an incorrect guess.
12. I need to check if the guess is a valid letter
13. If the user guesses a string congratulation message is displayed.
14. If the user runs out of guesses, the user is notified that they lost, what the secrets string was and what letters were missing from their guesses.
15. At the end of the game the user is asked if they want to repeat the game one more time.

Problem 6

DESING:

1. I got some ideas from Sean Harrington on the discussion board
2. Array of words need to be declared
3. To randomly pick number of words, Ill do srand(time(0)); words=rand()%(3-2+1) +2; //basically I am using a rand() % (max1 - min1+ 1) + min1. Minimum number of words will be 2 and maximum number of words will be
4. Next step will be to chose words from the array. Ill do awhile loop.
   while(wordNumber>=1 && wordNumber<=words)

```
{
   srand(time(0));
randomize (phrase_array, n);// randomize array, see
http://www.geeksforgeeks.org/shuffle-a-given-array/
 wordpick = rand()%SIZE;
temp=phrase_array[wordpick];
 phrase=phrase+temp;
        wordNumber++;
}
```

5. Phrase will be a string that needs to be gueesed
6. Declare variables: we need to keep track of misses, guesses, exposed letters, valid letters
7. cin >> to let the user make an entry.
8. If the entry is not a letter, let the user know that entry was invalid
9. Exposed letters will be incremented each time the letter is correctly guessed. This is critical.
10. Initialize: guesses = 3; This is a programmers choice. I decided that user can make only 3 INCORRECT guesses. Please note that I am counting only INCORRECT GUESSES.
11. Initialize string, lets call it display. This string will keep track of letter entered by user
12. At this point, we will initialize   string display = word;
13. Next step will be to replace all charactesr in the string word with '-'. I saw this code in the book

```
for (int i = 0; i < display.length(); i++)
        display [i] = '-';
```

14. Next step possibly   if stament is (guesses > 0) or possibly   while(exposed < word.length() && guesses > 0)

. Until exposed count is less than word. Length., continue this loop.
In this loop we need to let the user know how many guesses they have left.
How many missed they already made
Ask the user to enter a letter
Do cin to let the user enter a letter

15. Once the guess if made, we need to figure out if the guess if correct and if the guess is the duplicate (possibly 2 bool statement (goodguess= false, duplicate = false)
16. For each entry the user makes, we need to loop through the entry: for(int i=0; i < word.length(); i++) and compare the entry to the word: if(response == word[i])
17. For duplicate do something like this:
        if (display [i] == word [i]) . Let the user know that entry is duplicate, change bool to false

18.   If the entry is not duplicate: do else stamen: display[i] = word [i];

```
        a. Count  exposed++;
        b. Set the bool goodGuess =true; // guess was good, so we change bool
           to good
      if (!goodGuess)
     {
         misses++; // increment misses
          guesses--;// decrementing guesses
          // let the user know that the letter is not in the word
```

19. To update the list of valid entries do a for loop, declare an array of possible letters. Make a bool statement and set all indexes in the array set to true.

```
                    for (i=0; i<26; i++)
        {
                 if (alphabet_b[i]==true) //shows what letters the person has not
  guessed yet
                 {
```

If the letter was entered already, change the bool to false, and it will not be displayed to the user.

20. If there word is guessed : diplay congratulations message
21. If (guesses <=0) the user, ran out of guesses, let the user know that they lost, display the word and let them know which characters they have not guesses. I have no idea at this point how to display the characters that were missed
22. Do while loop to ask the player if they want to play one more time.

TESTING

testing	Input (tested conditions)	Expected output	Pass/Fail
Random number of words to be selected changes	cout statement to find out how many words are selected	Different # of words is selltected	pass
Random words are chosen and are put in the string	Cout statement to find out the string that needs to be guessed	Different string is chosen	pass
User	Guess the letter	The letter replaces '-' in the word	pass
User	Guesses the letter	Guessed letter is removed from the valid list of letters	pass
User	Letter is not guesses	User is notified that the letter is not in the word	Pass
User	Does not guess a letter	Entered letter is removed from the valid list of letters	pass
User	Letter is not guesses	Guesses – Misses ++	Pass

User	Letter that was guesses already, entered one more time	Lets the user know that letter was already guesses	Pass
User	Letter that was guesses already, entered one more time	Guesses and misses, stay the same	Pass
User	User runs out of guesses and does not guess the word	User is notified that they lost	Pass
User	User runs out of guesses and does not guess the word	Secret word if displayed to the user	Pass
User	User runs out of guesses and does not guess the word	Letter that were not guessed are displayed to the user	Pass
User	User guesses the work	Congratulations message is displayed	Pass
User	Runs out of guesses	Asks the user if they want to play again. If they say yes, game starts over	Pass
User	Runs out of guesses	Asks the user if they want to play again. If they say no, exits the screen	Pass

Refection:
I spent over 60 hours on the assignment this week. Reflection…. Its was hard that all I say because if I start writing something it will take at least 10 pages.