MEDGAR EVERS
COLLEGE
The City University of New York

Data Structures
CS 246 - ON40
Department of Physics and Computer Science
Medgar Evers College
Exam 3

Direction: Modify the `"exam03.cpp"` file provided; you cannot include additional libraries. Afterwards, submit your typed work in the Exams directory of your github repository and/or as an attachment on Google classroom under the Exam03 assessment. All submissions should have their appropriate extensions.

Problem	Maximum Points	Points Earned
1	5	
2	5	
3	5	
4	5	
Total	20	

1. Write the definition of the function `SelectionSort()` whose header is

```
template <typename T>
void SelectionSort(Node<T>* root)
```

Given that *root* is referencing a doubly linked list, it sorts the list using the selection sort method in ascending order. It must sort the data of the linked list; not the nodes of the linked list.

2. Write the definition of the function `AdjacentDuplicateRemoval()` whose header is

```
string AdjacentDuplicateRemoval(string str)
```

It returns a reduce form of the *str* that has no adjacent duplicates. A string has an adjacent duplicate if two adjacent characters are equal. For instance, The function calls `AdjacentDuplicateRemoval("abbaca")` and `AdjacentDuplicateRemoval("passse")` will evaluate to `"ca"` and `"pase"` respectively.

3. The correlation of the indices of a two-dimensional array for a chess board and a one-dimensional array are as follows

$r \backslash c$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

where the border numbers are the row and column indices of the two-dimensional array and the numbers enclosed in the squares are the correlating indices of the one-dimensional array. Hence,

$$i = 8 * r + c$$

where i is the one-dimensional index, r is the row index of the two-dimensional array and c is the column index of the two-dimensional array.

Write the definition of the function `QueenPath()` whose header is

```
bool QueenPath(char bd[],int s,int e)
```

Given that *bd* represents a chess board that consists only of the characters `'o'` for occupied space and `'x'` for free space, the function returns true if a queen whose start position is s can make it to the end position e in any number of steps if both s and e are valid indices of *bd*; otherwise, it returns false. It does not matter what the characters are for the start and end spaces; however, movement to occupied spaces are prohibited. Recall that a queen can move horizontally, vertically and diagonally.

Hint: Keep track of positions visited. Likewise, be cautious about left and right adjacent moves.

4. For the function call `V("abbaaaabbaaabbaabbaabba")` construct an array trace table (or list) of the stack object where the function `V()` is defined as follows

```
bool V(string str)
{
 Stack<int> s;

 for(int i = 0;str[i] != '\0'; i += 1)
 {
  if(tolower(str[i]) == 'a')
  {
   if(s.IsEmpty() || s.Top() == 2)
   {
    s.Push(0);
   }
  }
  else if(tolower(str[i]) == 'b')
  {
   if(s.IsEmpty() || s.Top() == 2)
   {
    return false;
   }
   else if(s.Top() == 0)
   {
    s.Push(1);
   }
   else
   {
    s.Pop();
    s.Push(2);
   }
  }
  else
  {
   return false;
  }
 }
 return (s.IsEmpty() || s.Top() == 0);
}
```