Week 10

Remind

- 抄襲一律 0 分 （包含被抄襲者）
- 繳交期限: **11/22(Sun.) 11:59 p.m.**
- 繳交的檔案格式、名稱請符合以下規定
    - 請繳交 zip檔至 Ceiba作業區，名稱為 <student_id>.zip
    - 解壓縮後須符合格式、名稱
    - e.g. b12345678.zip
- 必須完成 Demo 才可以提早離開
- 若沒有完成 Demo 就中途早退，視同缺席

Problem 1 - Palindrome (0.5%)

Description

A string is a palindrome if it is the same as its reverse order. Given a string, determine whether it is a palindrome.

Input

A string which length will not exceed 100.

Output

"true" if it is a palindrome, "false" otherwise.

Sample input

```
abcba
```

Sample output

```
true
```

File name

{Student_ID}_1.cpp

Problem 2 - Text editor (2%)

Description

In this assignment, you are asked to implement a text editor.

Like your normal text editor, it can not only show your input from keyboard but can also process some slightly complicated operations as well.

For the text editor you should implement, it should be able to handle all these operations:

1. Character inputs from keyboard. (Will not contain any upper case letters)
2. Backspace (denoted by a '\b' in the input string)
3. Cursor moving (denoted by a '\l' for moving cursor one character to the left, '\r' for the other direction)
4. Shift (denoted by '\S' and '\s'. Note that '\S' with a capital S implies PRESSING the shift button, '\s' implies RELEASING the shift button. In other words, everything enclosed between a '\S' and a '\s' is pressed while holding the shift button.)
5. Copy (denoted by '\c', can be used in combination with '\l' and '\r' when shift is toggled ON)
6. Paste (denoted by '\v'. Paste everything you have copied to where your cursor is)
7. New line (denoted by '\n')
8. Upper case transformation (when shift is toggled ON, lower case letter should be transformed to upper case letter.)

Input

The keyboard input sequence.

The key board input will not exceed 500 characters.

Output

The text content.

The text content will not exceed 500 characters.

Sample input

```
hey\b\b\byo \S\l\l\l\c\s\r\v\v\vc\b\Sc\sindy what\S\l\l\l\l\c\s\r? \v's wrong wit
h me...
```

Sample output

```
yo yo yo yo Cindy what? what's wrong with me...
```

File name

{Student_ID}_2.cpp

Problem 3 - Select rank (1.5%)

3a. Array Permutation (0.25%)

`void randomPermArray(int data[], int n)`

Algorithm

1. Randomly pick an element from data[0] to data[n-1], swap it with data[n-1].
2. Decrement n by 1.
3. Repeat (Step.1 + Step.2) until every element in data is permuted.

Sample Input

```
arr[10] = {6,8,10,4,2,3,5,9,7,1};
randomPermArray(arr, 10);
for (int i=0; i<size; i++){
  cout << arr[i] << ' ';
  }
```

Sample Output

```
7 5 4 1 3 6 9 10 8 2
```

File Name

```
3a.cpp
```

3b. Binomial Estimation (0.5%)

`void testRandomPermArray(int n, double p, double& mean, double& var)`

Algorithm

1. Create an array of size 10000, initialize the first floor(10000*p) elements to 1 and the remaining to 0.
2. Use `randomPermArray` you implemented in problem 3a. to permute the array.
3. Repeat (Step.1 + Step.2) total of n times. Record the number of 1s that appear in each position (0~9999) of the array . Divide that number by n, and now you have 10000 numbers (0~1).
4. Calculate the mean and variance of of the 10000 numbers, and put them in `mean` and `var` . If you did everything right, this should follow the binomial distribution: `mean` is close to p; `var` is close to $\frac{p(1-p)}{n}$.

Sample input

```
testRandomPermArray(100, 0.1, mean, var);
printf("%f %f",mean,var);
```

Sample output

```
0.100000 0.000899
```

File Name

```
3b.cpp
```

3c. Find `rank` th smallest data in array (0.75%)

For this particular problem, you will be implementing 3 functions.

Function 3c_1

`int selectRank(int data[], int n, int rank)`

This function reads an array, its left boundary, right boundary, and returns the rankth smallest element from data[left] to data[right].

- Input: `data[]` is an array of size n
- Output: rankth smallest element of `data[]`

You may use the algorithm given below to complete this task.

1. Call `partitionArray` with pivot being the leftmost element of the array.
2. Once the array is partitioned, you know that the rankth smallest is either
    1. the pivot
    2. the left subarray
    3. the right subarray
3. If the pivot is the rankth smallest element, return it. Else, recursively call `selectRank` and narrow left and right indices until the rankth element is found.

To do this, write the following auxiliary functions.

Function 3c_2

`some_type partitionArray()`

You should be familiar with this partition function, since you have already implemented it in week 8.

Function 3c_3

`int selectRank(int data[], int left, int right, int rank)`

- Input: `data[]` is an array (now you don’t need the size info)
- Output: rankth smallest element from data[left] to data[right]

File Name

```
3c.cpp
```