

Lab Quiz 2 CS111 Summer 2020

NAME _____ CUNYID _____

Instructions :

- 1) Make sure your name and CUNY ID are filled in.
- 2) When asked to write a program, begin with the main portion of the program. In most cases I have written the beginning of the function for you and you just fill in the function body.
- 3) The only outside functions you can use are :
 - a) cout, cin, rand(), srand(), and time()
 - b) functions you write yourself
 - c) functions that appear anywhere on this test
 - d) Each question is worth 10 points each.
 - e) you cannot use the string object

Q1) FillFibonacciArray

Sample Output :

```
1 1 2 3 5 8 13 21 34 55
1 1 2 3 5 8 13 21 34 55
2 2 4 6 10 16 26 42 68 110
3 3 6 9 15 24 39 63 102 165
5 5 10 15 25 40 65 105 170 275
8 8 16 24 40 64 104 168 272 440
13 13 26 39 65 104 169 273 442 715
21 21 42 63 105 168 273 441 714 1155
34 34 68 102 170 272 442 714 1156 1870
55 55 110 165 275 440 715 1155 1870 3025
```

```
*/
```

```
#define SIZE 10
```

```
/*
```

1) initialize the array with 2 rows and 2 columns equal to 1.

i.e. 1 1

```
11
```

2) add across for each column where each column is equal to the two previous columns added together

i.e. 1 1 2 3 5 8 13 21 34 55

3) then multiply down where each column in each row is equal to the to the sum of two columns above it.

4) Use for loops for to fill the arrays. Do not do so by hand. SIZE can be any value from 2..100. In this example it

is 10.

```
*/
```

```
void FillFibonacciArray(int arr[SIZE][SIZE])
```

```
{
```

```
    for (int i = 0; i < 2; i++)
```

```
        for (int j = 0; j < 2; j++)
```

```
            arr[i][j] = 1;
```

```

for (int i = 0; i < 2; i++)
    for (int j = 2; j < SIZE; j++)
        arr[i][j] = arr[i][j - 1] + arr[i][j - 2];
for (int i = 2; i < SIZE; i++)
    for (int j = 0; j < SIZE; j++)
        arr[i][j] = arr[i - 1][j] + arr[i - 2][j];
for (int i = 0; i < SIZE; i++)
{
    for (int j = 0; j < SIZE; j++)
        cout << arr[i][j] << " ";
    cout << endl;
}
}
/*

```

Q2)

RandomizeString - Randomly move each letter in a string making sure it doesn't directly go to the same spot

Note : A character can be in the same place as the original character but cannot be moved there directly. E.g. if 'C' is in position 2, 'C' cannot be moved directly to position 2, but it can be moved to position three and from there back to position 2.

RandomizeString("ABC")?"BCA"

*/

```

void Swap(char& c1, char& c2)

```

```

{
    char temp = c1;
    c1 = c2;
    c2 = temp;
}

```

```

int StringLength(const char* str)

```

```

{
    int i = 0;
    while (str[i])
        i++;
    return(i);
}

```

```

char* RandomizeString(char* str)

```

```

{
    for (int i = 0; i < StringLength(str); i++)
    {
        int r = i;
        while (r == i)
            r = rand() % StringLength(str);
        Swap(str[i], str[r]);
    }
}

```

```

        return(str);
    }
/*Q3) SpaceMyself – Prints str with a space after each string.
There is one extra space after each letter.
SpaceMyself("ABCDEF")? "A B C D E F"
SpaceMyself("AB")? "A B"
*/
void SpaceMyself(const char* str)
{
    for (int i = 0; i < StringLength(str); i++)
    {
        cout << str[i];
        int j = i;
        //while (j >= 0)
        //{
        //    cout << " ";
        //    j--;
        //}
        do
        {
            j++;
            cout << " ";
        } while (j < i);
    }
}
/*

```

Q4) FillMultiplicationArray

Sample Output :

```

1 2 2 4 8
1 2 2 4 8
1 4 4 16 64
1 8 8 64 512
1 32 32 1024 32768

```

```
#define SIZE 5
```

1) initialize the array with 2 rows and 2 columns equal to 1 and 2.

i.e. 12

```
12
```

2) multiply across for each column where each column is equal to the two previous columns multiplied together

i.e. 1 2 2 4 8

3) then multiply down where each column in each row is equal to the to the multiple of two rows above it.

4) Use for loops for to fill the arrays.Do not do so by hand.SIZE can be any value from 2..100.In this example it

is 5.

```
*/  
#define ARR_SIZE 5  
void FillMultiplicationArray(int arr[ARR_SIZE][ARR_SIZE])  
{  
    for (int i = 0; i < 2; i++)  
        for (int j = 0; j < 2; j++)  
            arr[i][j] = j + 1;  
    for (int i = 0; i < 2; i++)  
        for (int j = 2; j < ARR_SIZE; j++)  
            arr[i][j] = arr[i][j - 1] * arr[i][j - 2];  
    // start from row 2  
    for (int i = 2; i < ARR_SIZE; i++)  
        for (int j = 0; j < ARR_SIZE; j++)  
            arr[i][j] = arr[i - 1][j] * arr[i - 2][j];  
    for (int i = 0; i < ARR_SIZE; i++)  
    {  
        for (int j = 0; j < ARR_SIZE; j++)  
            cout << arr[i][j] << " ";  
        cout << endl;  
    }  
}
```

```
/*
```

Q5) OrderInAlphabet

```
int OrderInAlphabet('b')?2
```

```
int OrderInAlphabet('B')?2
```

For upper or lower case characters, returns the order they appear in the alphabet

```
*/
```

```
int OrderInAlphabet(const char c)
```

```
{  
    return(c >= 'A' && c <= 'Z') ? c - 'A' + 1 : c - 'a' + 1;  
}
```

```
/*
```

Q6) DuplicateMyself

Duplicates whatever value I am in the array backwards

```
DuplicateMyself("ABC")? "ABCCBA"
```

```
DuplicateMyself("AB")? "ABBA"
```

```
*/
```

```
char* DuplicateMyself(char* str)
```

```
{  
    int dest = StringLength(str);  
    int src = StringLength(str) - 1;  
    while (src >= 0)  
    {
```

```
        str[dest] = str[src];
        dest++;
        src--;
    }
    str[dest] = 0;
    return(str);
}
int main()
{
    srand(time(0));
    int arr[SIZE][SIZE];
    FillFibonacciArray(arr);
    int arr1[ARR_SIZE][ARR_SIZE];
    FillMultiplicationArray(arr1);
    char str[10] = "ABCD";
    cout << RandomizeString(str) << endl;
    SpaceMyself("ABCDEF");
    cout << OrderInAlphabet('C') << endl << OrderInAlphabet('c') << endl;
    return(0);
}
```