

**BBM201 – Data Structures – Fall 2016**  
**1st Midterm**  
**11.11.2016**

Name Surname: \_\_\_\_\_

Student ID : \_\_\_\_\_ Section: \_\_\_\_\_

Süre ... dakikadır.

Question	1	2	3	4	5	6	7	8	Total
Points									100
Grade									

**Question 1.** Decide if the following statements are true or false, circle your answer. Give a short explanation if you chose 'False'.

a) A data structure is a way to store and organize data in computer, so that it can be used efficiently.

True

False

b)  $3n^2 + 10n \log n = O(n \log n)$

True

False

b) Big-Oh ( $O$ ) notation gives a lower bound on the running time of a program.

True

False

c) The cost of insertion (push) and deletion (pop) of an element in a stack is  $O(1)$ .

True

False

d) The complexity of the worst-case running time of binary search for an array of size  $n$  is  $n^2$ .

True

False

**Question 2.** A palindromic word is a sequence of characters that reads the same backward and forward. For example; *repaper*, *refer*, *kayak* are palindromic words. The method `IsPalindrome(char[] str)` checks if a given string is palindrome by using a stack. According to the given prototypes for stack operations, please fill in the gaps in the code.

```
void push(char);
void pop();

void IsPalindrome(char str[]){

    int len = strlen(str);
    for (i = 0; i < len; i++)
        _____;

    for (i = 0; i < len; i++)
        if (str[i] == _____)
            count++;

    if (_____)
        printf("%s is a palindromic string\n", str);
    else
        printf("%s is not a palindromic string\n", str);
}
```

### Question 3.

Please write the output of the following method.

```
void recursiveFun(int value)
{
    if(0 < value && value < 10)
    {
        recursiveFun(value - 2);
        recursiveFun(value + 1);
        printf(" %d", value);
    }
}
```

### Question 4.

Please write the output of the following method.

```
static int negative(int num)
{
    if(num >= 20)
        return -5;
    else
        return negative(num + 4) + 2 * num;
}
```

**Question 5.**

Let  $a[n][n]$  be an upper triangular matrix (see the example given below). The elements of this triangular matrix are stored in a one-dimensional array as given below:

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$
0	$a_{11}$	$a_{12}$	$a_{13}$
0	0	$a_{22}$	$a_{23}$
0	0	0	$a_{33}$

$U$

$a_{00}$	$a_{10}$	$a_{11}$	$a_{20}$	$a_{21}$	$a_{22}$	$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Please complete the method `readtriangularmatrix(int[], int)` that reads integers from the keyboard and fills the one-dimensional array  $U$  with an upper triangular matrix.

```
void readtriangularmatrix(int U[], int n){
    int i, j, k;
    if(n*(n+1)/2 > MAX_SIZE){
        printf("\n invalid array size \n");
        exit(-1);
    }
    else
        for(i=0; i<=n-1; i++){
            k=.....
            for(j=0; j<=i; j++)
                scanf("%d", .....);
        }
}
```

**Question 6.**

If  $C$  is the array shown with its address above each node, write what the following lines of a program will print in the empty column.

100		116		132							
3	4	1	5	0	2	9	8	7	2	10	12

`int C[3][2][2];`

The code	will print:
----------	-------------

<code>printf("C=%d", C);</code>	
<code>printf("C+2=%d", C+2);</code>	
<code>printf("**(*C+1) =%d", *(*C+1));</code>	
<code>printf(" *(C[0]+1)=%d '", *(C[0]+1) );</code>	
<code>printf(" *(C[2][1]+1)=%d '", *(C[2][1]+1));</code>	
<code>printf(" *((*(*C+2))+1)=%d '", *((*(*C+2))+1));</code>	