



# Introduction to Computing

CS 151 - 040

Department of Physical and Computer Sciences

Medgar Evers College

## Exam 4 - Take Home

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

**Directions:** Read the questions carefully. Write legibly to earn credit.  
**Good Luck!**

Section	Max Points	Points Earned
1	8	
2	8	
3	4	
<b>Total</b>	20	

## Section 1: Syntax

Write **ONLY** the statements requested and required.

- 1) Initialize three char arrays to “Initial” in different ways. (Initialization is the process of declaring and assigning a value to a variable in a single step.)
- 2) Write the declaration of a struct named **Grade** that has a double and a string as members named *gpa* and *letter* respectively.
- 3) Given a double pointer variable, *e*, that has been declared, write a statement(s) that perform a single dynamic memory allocation and assigns it 2.71828.
- 4) Given a double array, *num*, that has been initialized consisting of 10 elements and a double pointer variable, *ptr*, that has been declared, write a statement(s) that assigns the address of *num* to *ptr*, and then, double each element of *num* using only *ptr* (you cannot use *num* in the assignments).
- 5) Given a string pointer variable, *str*, that has been dynamically allocated for a block, write a statement(s) that will deallocate *str*.
- 6) Given an int pointer variable, *t*, that has been declared, write a statement(s) that performs a block dynamic memory allocation of 15 and assigns it the first 15 positive even integers (2 is the first number).
- 7) Given an int, *n*, that has been initialized and an int pointer variable, *ptr*, that has been declared, write a statement(s) that assigns the address of *n* to *ptr*, and then, assign 3 more than the four times *n* to *n* using only *ptr* (you cannot use *n* in the assignment).
- 8) Given a char pointer variable, *ch*, that has been dynamically allocated for a single element, write a statement(s) that will deallocate *ch*.

## Section 2: Program

Write the complete program. Use "Stack.h"; however, do not access the members of the Stack or Node structs directly.

```
#ifndef STACK_H
#define STACK_H

struct Node
{
    char data;
    Node* link;
};

Node* createNode(char data, Node* link)
{
    Node* tmp = new Node;
    tmp->data = data;
    tmp->link = link;
    return tmp;
}

struct Stack
{
    Node* top;
};

void initialize(Stack& _this)
{
    _this.top = NULL;
}

bool isEmpty(const Stack& _this)
{
    return _this.top == NULL;
}

char top(const Stack& _this)
{
    return _this.top->data;
}

void push(Stack& _this, char item)
{
    Node* tmp = createNode(item, _this.top);
    _this.top = tmp;
}

void pop(Stack& _this)
{
    if(isEmpty(_this))
    {
        Node* tmp = _this.top;
        _this.top = _this.top->link;
        delete tmp;
        tmp = NULL;
    }
}

#endif
```

Define the function

<b>Function Name:</b>	<b>IsPalindrome()</b>
<b>Parameter(s):</b>	<i>str</i> : char*
<b>Return:</b>	bool

where **IsPalindrome()** returns true if *str* is a palindrome; otherwise, it returns false. However, you must use only a single Stack object.

In the main function,

1. declare a char array of at least 512 elements.
2. prompt the user to enter a string and store it in the char array.
3. call **IsPalindrome()** with the char array as the argument.
4. display the output of **IsPalindrome()** as boolean string (true or false).

### Section 3: Extra Credit

Write the function definition of the bool function named `PushDownAuto` that takes a c string named *str* as a parameter. It returns true if *str* is in the format  $A^n C^m B^{2n}$  where  $n \geq 0$  and  $m > 0$ ; otherwise, it returns false. Other characters are not allowed.