

**Carleton University**  
**Department of Systems and Computer Engineering**  
**SYSC 2006 B - Foundations of Imperative Programming - Fall 2013**

**Midterm Exam Answers - October 17, 2013**

**Question 1**

```
#include <stdio.h>

int main(void) {
    int i, j, n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);

    for(i=0; i<n; i++) {
        for(j=0; j<n-1-i; j++)
            printf(" ");
        for(j=0; j<=i; j++) {
            int r =factorial(i)/(factorial(j)*factorial(i-j));
            printf("%d ", r);
        }
        printf("\n");
    }
    return 0;
}
```

## Question 2

a)

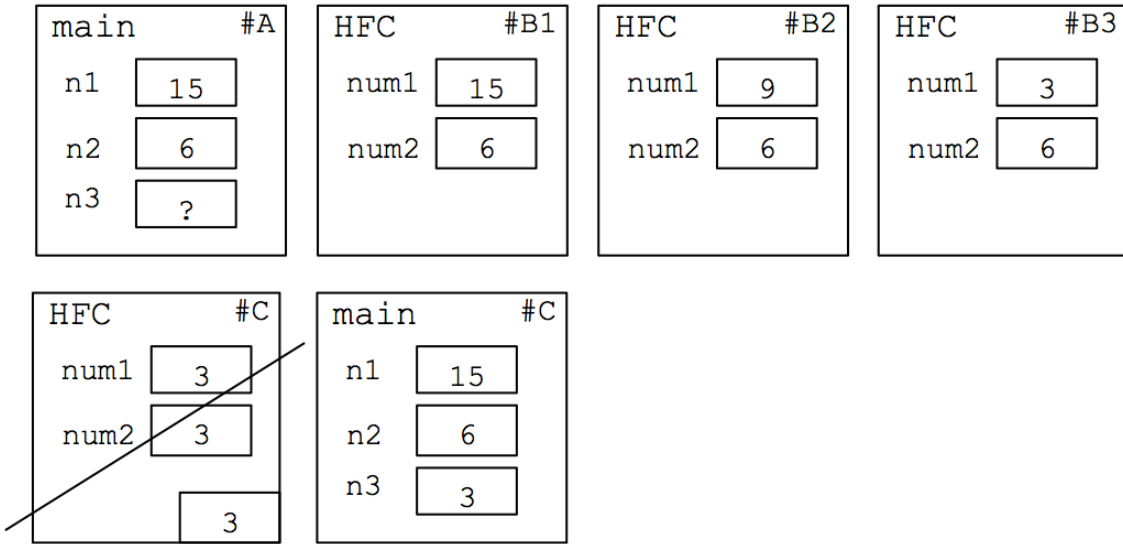
```
void toTitleCase(char s[]) {
    int i;
    for(i=0; s[i]!='\0'; i++)
        if (i==0 || s[i-1]==' ')
            if (s[i]>='a' && s[i]<= 'z')
                s[i] -= 32;
}
```

b)

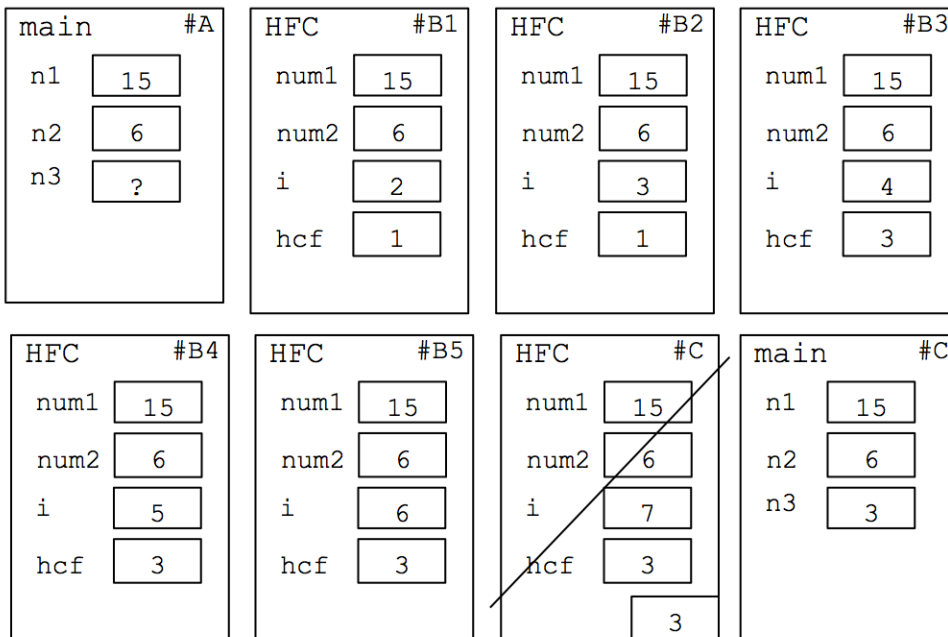
```
void testTitleCase(char s[], const char t[]) {
    printf("Original: %s\n", s);
    toTitleCase(s);
    printf("Converted: %s\n", s);
    printf("Expected: %s\n", t);
    if (strcmp(s, t)==0)
        printf("Test: Success\n");
    else
        printf("Test: Failure\n");
}
```

### Question 3

a)



b)



**Notice:** the “B” stack frames above should have gone until “i=15” since the condition in the for loop had a “||” (vs. the intended “&&”), so those who gone that far took a few bonus points.

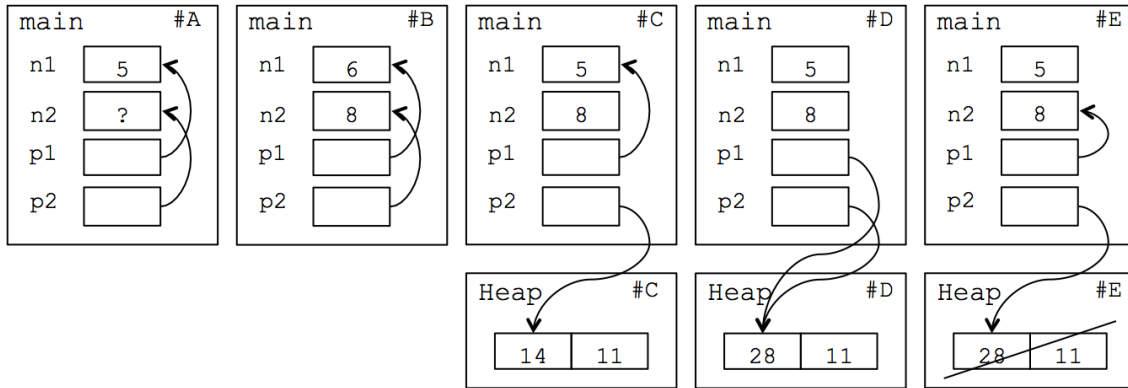
c) The implementation in (a) is better because:

- 1) It is faster (less iterations in the loop)

2) It uses less memory (does not define any local variables).

## Question 4

a)



b) `p1` now points to a local variable `n2` on the stack, hence freeing it causes undefined behavior.

c) `p1` now points to deallocated memory (same as the freed `p2` pointer), hence writing to it causes undefined behavior

d) `p2` now points to a single integer allocated on the heap, hence writing to the next unallocated integer on the heap causes undefined behavior.

e) `p2` is now the only pointer to the allocated integer on the heap. As `p2` is overridden by an unrelated address (that in `p1`), this leads to a memory leak since the allocated integer on the heap cannot consequently be freed.