

Exam 8 - Problem 2



Exam 8 - Problem 2



Given the following function written in C:

```
int compute(int M[10][10], int m, int n)
{   int i, sum, row;
    sum = 0;
    row = 0;
    for (i = m; i < n; i++)
        sum = sum + Stratify(M[row][i], &row);
    return (sum + 1);
}
```



Exam 8 - Problem 2

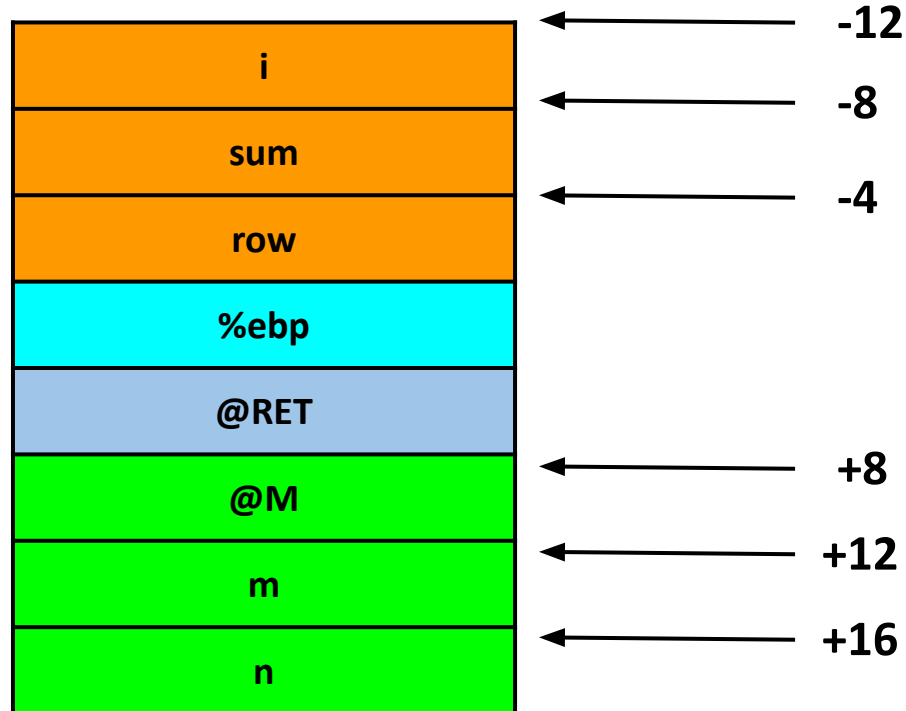


- a) Draw the activation block of the function.
- b) Translate the function to x86 assembler.
Suppose the Stratify function is already programmed.

Exam 8 - Problem 2



a) Draw the activation block of the function.



Exam 8 - Problem 2



Part 1/3

```
compute:
  pushl %ebp
  movl %esp, %ebp
  subl $12, %esp
  pushl %ebx
  movl $0, -8(%ebp)      ; sum = 0
  movl $0, -4(%ebp)      ; row = 0
  movl 12(%ebp), %ebx     ; %ebx = m
```

Exam 8 - Problem 2



Part 2/3

for:

<code>cmpl 16(%ebp), %ebx</code>	<code>; compare i with n</code>
<code>jge endfor</code>	<code>; if i >= n jump to endfor</code>
<code>leal -4(%ebp), %eax</code>	<code>; %eax = &row</code>
<code>pushl %eax</code>	<code>; push &row to pass as argument</code>
<code>movl -4(%ebp), %edx</code>	<code>; %edx = row</code>
<code>imull \$10, %edx</code>	<code>; %edx = 10 * row</code>
<code>addl %ebx, %edx</code>	<code>; %edx = 10 * row + i</code>
<code>movl 8(%ebp), %ecx</code>	<code>; %ecx = @M</code>
<code>movl (%ecx, %edx, 4), %edx</code>	<code>; %edx = M[row][i]</code>
<code>pushl %edx</code>	<code>; push M[row][i] to pass as argument</code>

x86

Exam 8 - Problem 2



Part 3/3

```
call Stratify
addl $8, %esp           ; we have pushed 8 bytes
addl %eax, -8(%ebp)     ; sum = sum + Stratify(M[row][i],&row)
incl %ebx              ; i++
jmp for
endfor:
movl -8(%ebp), %eax     ; %eax = sum
incl %eax              ; %eax = sum + 1
popl %ebx              ; pop register %ebx
movl %ebp, %esp
popl %ebp
ret
```

x86