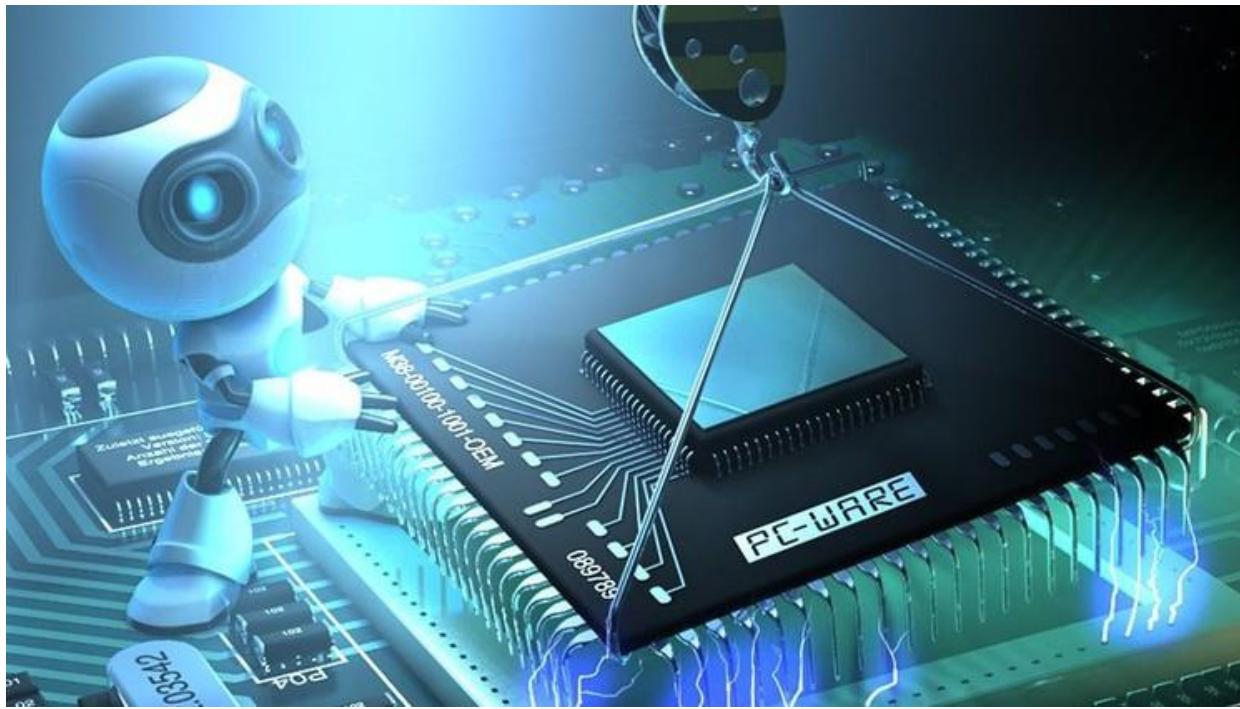


Exam 8 - Problem 2



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

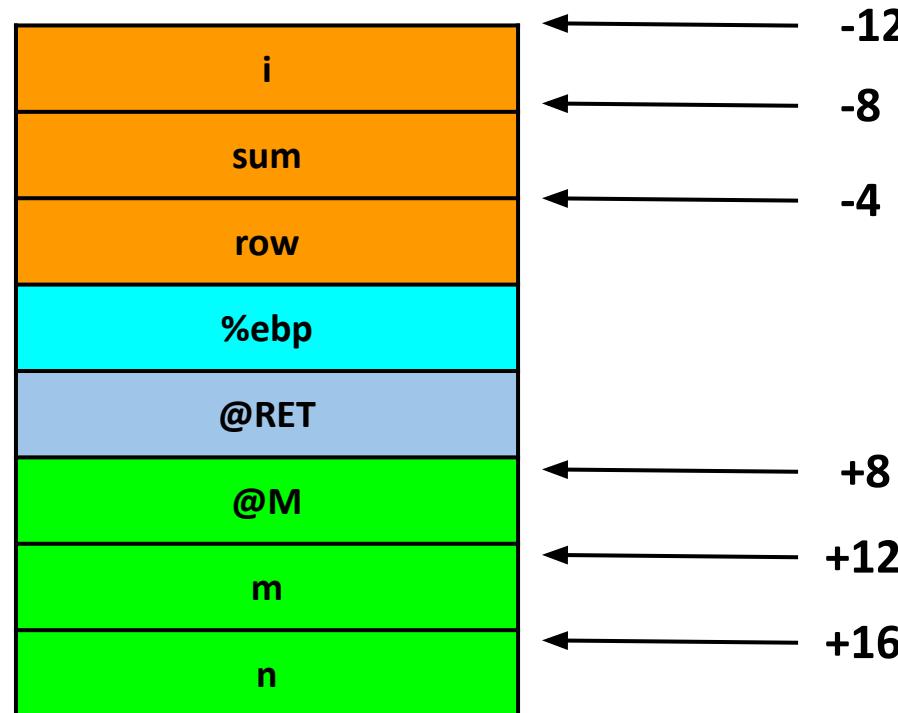


- a) Draw the activation block of the function.
- b) Translate the function to x86 assembler.
Suppose the Stratify function is already programmed.

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:

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

x86

Exam 8 - Problem 2



Part 3/3

```
call Stratify
addl $8, %esp
addl %eax, -8(%ebp)      ; we have pushed 8 bytes
incl %ebx
incl %eax                ; sum = sum + Stratify(M[row][i],&row)
                           ; 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