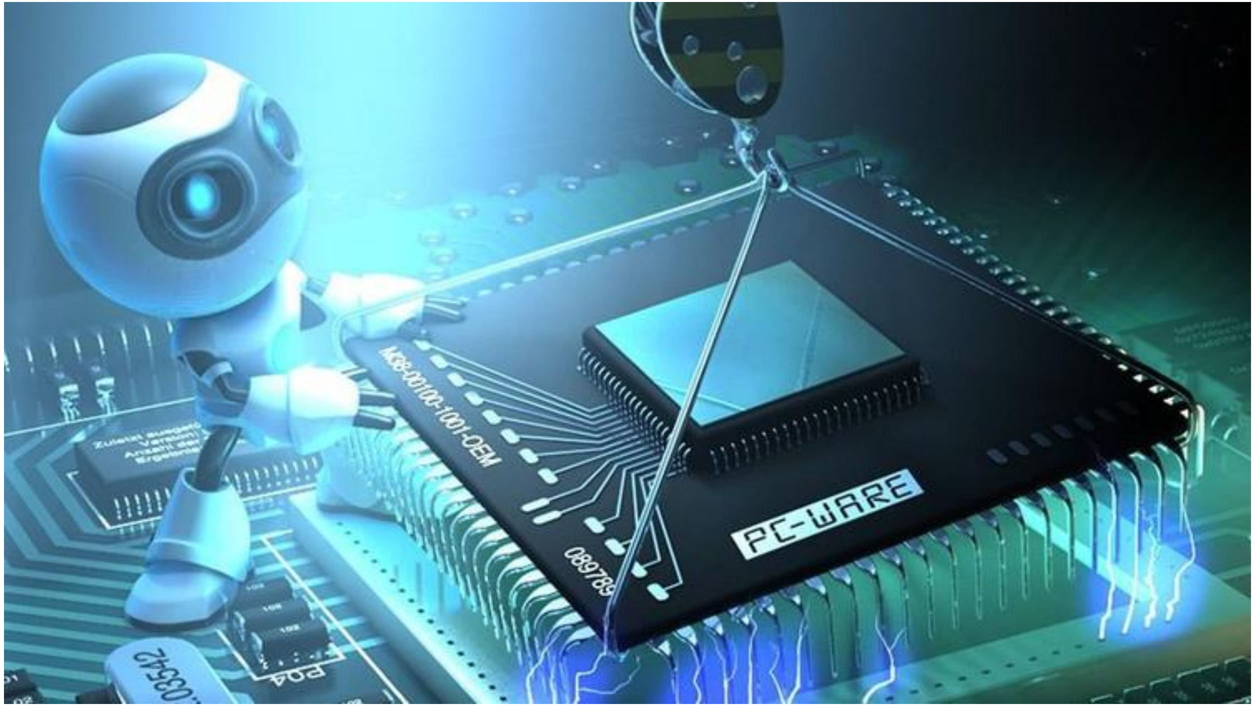
Exam 9 - Problem 1



Exam 9 - Problem 1

Given the following function written in C:

```c
int maximum (int v[300], int i)
{  int res;
   if (i == 0) res = v[0];
   else {
      res = maximum(v, i-1);
      if (v[i] > res)
         res = v[i];
   }
   return res;
}
```

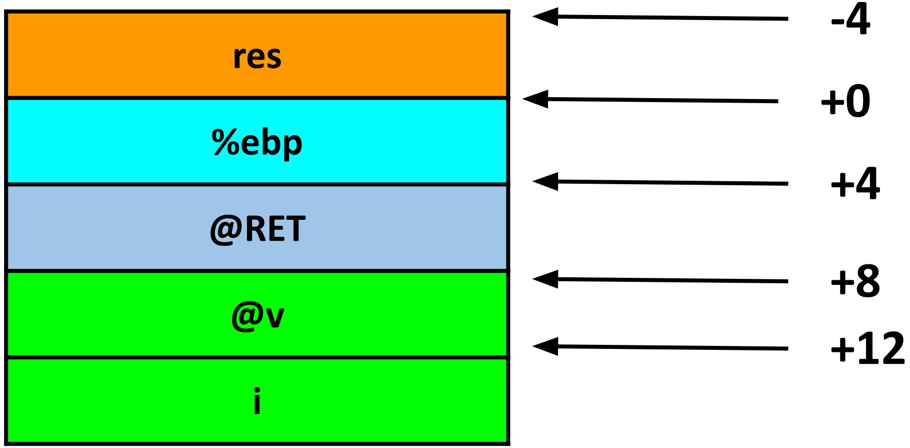
Exam 9 - Problem 1

a) Draw the activation block of the function.

b) Translate the function to x86 assembler.

Exam 9 - Problem 1

a)



res

%ebp

@RET

@v

i

-4

+0

+4

+8

+12

Exam 9 - Problem 1

b)

```
maximum:
    pushl   %ebp
    movl    %esp, %ebp
    subl    $4, %esp                ; We have only 1 local variable (integer)

    movl    12(%ebp), %ecx          ; Load the value of i into %ecx
    cmpl    $0, %ecx                ; Compare i with 0
    jne     else_condition          ; Jump to else_condition if i != 0

    movl    8(%ebp), %eax           ; Load the value of v[0] into %eax
    jmp     end                     ; Jump to the end of the function
```

Part 1/3

Exam 9 - Problem 1

b)

```
else_condition:
   movl    12(%ebp), %edx      ; Load the value of i into %edx
   subl    $1, %edx            ; %edx = i - 1
   pushl   %edx                ; Push i-1 onto the stack
   pushl   8(%ebp)             ; Push v onto the stack
   call    maximum             ; Recursive call to maximum function
   addl    $8, %esp            ; Clean up the stack after the recursive call

   movl    8(%ebp), %edx       ; %edx = @v
   movl    12(%ebp), %ebx      ; %ebx = i
   imull   $4, %ebx            ; Multiply i by 4 to calculate the offset
   add     %ebx, %edx          ; %edx = @v[i]
   movl    (%edx), %edx        ; %edx = v[i]
```

Part 2/3

Exam 9 - Problem 1

b)

```
    cmpl    %eax, %edx              ; Compare v[i] with res
    jle     end                     ; Jump to end if v[i] <= res

    movl    %edx, %eax              ; Move v[i] into %eax as the new res

end:
    addl $4, %esp
    movl %ebp, %esp
    popl %ebp
    ret
```

Part 3/3