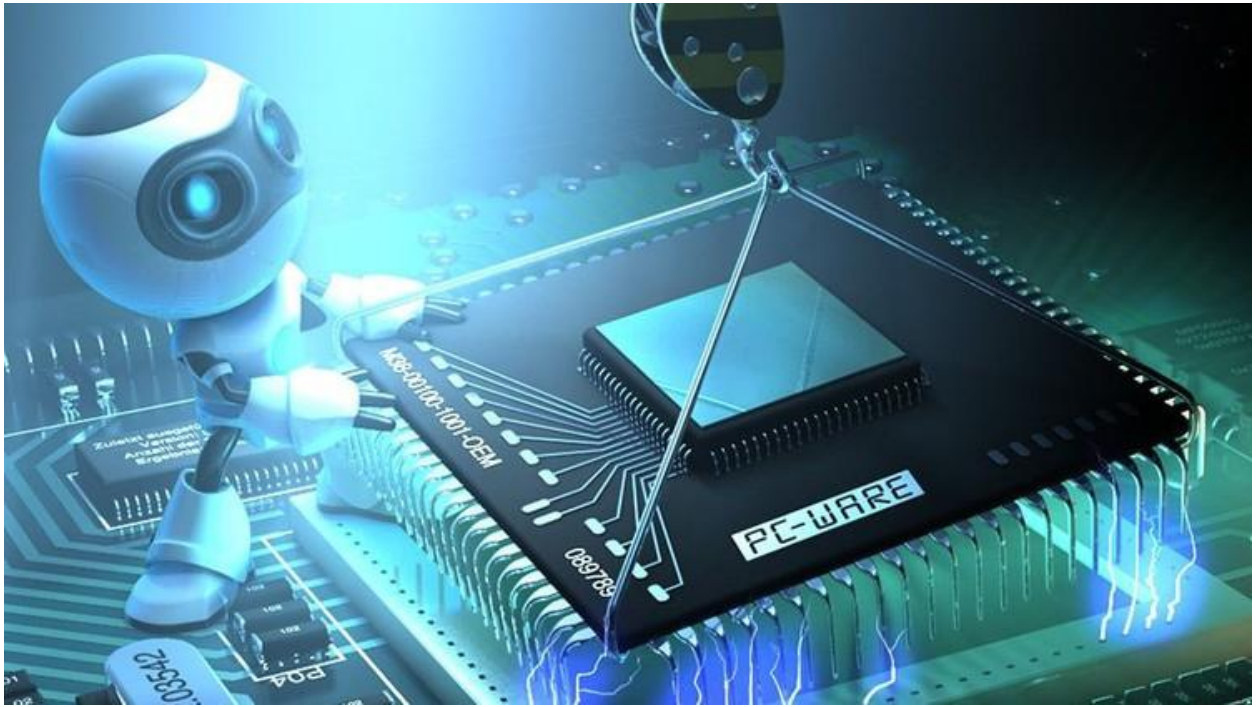
Exam 9 - Problem 2



Exam 9 - Problem 2

Given the following function written in C:

```
void exam (int a, int b[100], int *c)
{ int d[100];
 int aux;
 ...
}
```

Exam 9 - Problem 2

a) Draw the activation block of the function.

Translate the following statements into x86 assembler assuming they are in the body of the exam subroutine:

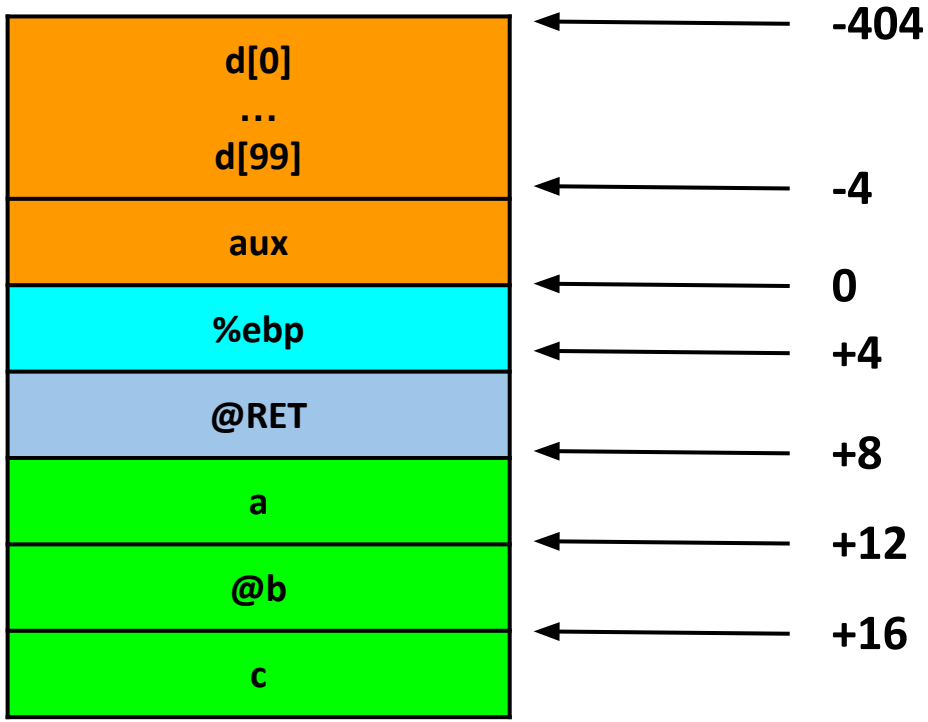
b)
```
exam(0, d, &aux);
```

c)
```
for (aux = 0; aux < 100; aux++)
    b[aux] = d[aux];
```

d)
```
exam(a, b, c);
```

Exam 9 - Problem 2

a)



Exam 9 - Problem 2

b) `exam(0, d, &aux);`

```
leal -4(%ebp), %eax          ; %eax = &aux
leal -404(%ebp), %ecx        ; %ecx = &d
pushl %eax                   ; we pass the argument &aux
pushl %ecx                   ; we pass the argument &d
pushl $0                     ; we pass the argument 0
call exam
```

Exam 9 - Problem 2

c)
```
for (aux = 0; aux < 100; aux++)
    b[aux] = d[aux];
```

```
movl $0, %ecx                   ; ecx represents aux (initially = 0)
for: cmpl $100, %ecx
jge endfor                      ; if aux >= 100 jump to endfor label
leal -404(%ebp), %eax           ; %eax = @d
movl (%eax, %ecx, 4), %eax      ; %eax = @d + aux*4  = d[aux]
movl 12(%ebp), %edx             ; %edx = @b
movl %eax, (%edx, %ecx, 4)      ; b[aux] = d[aux]
incl %ecx                       ; aux++
jmp for
endfor:
```

Exam 9 - Problem 2

d) `exam(a, b, c);`

```
pushl 16(%ebp)        ; we pass the argument c
pushl 12(%ebp)        ; we pass the argument b
pushl 8(%ebp)         ; we pass the argument a
call exam
```