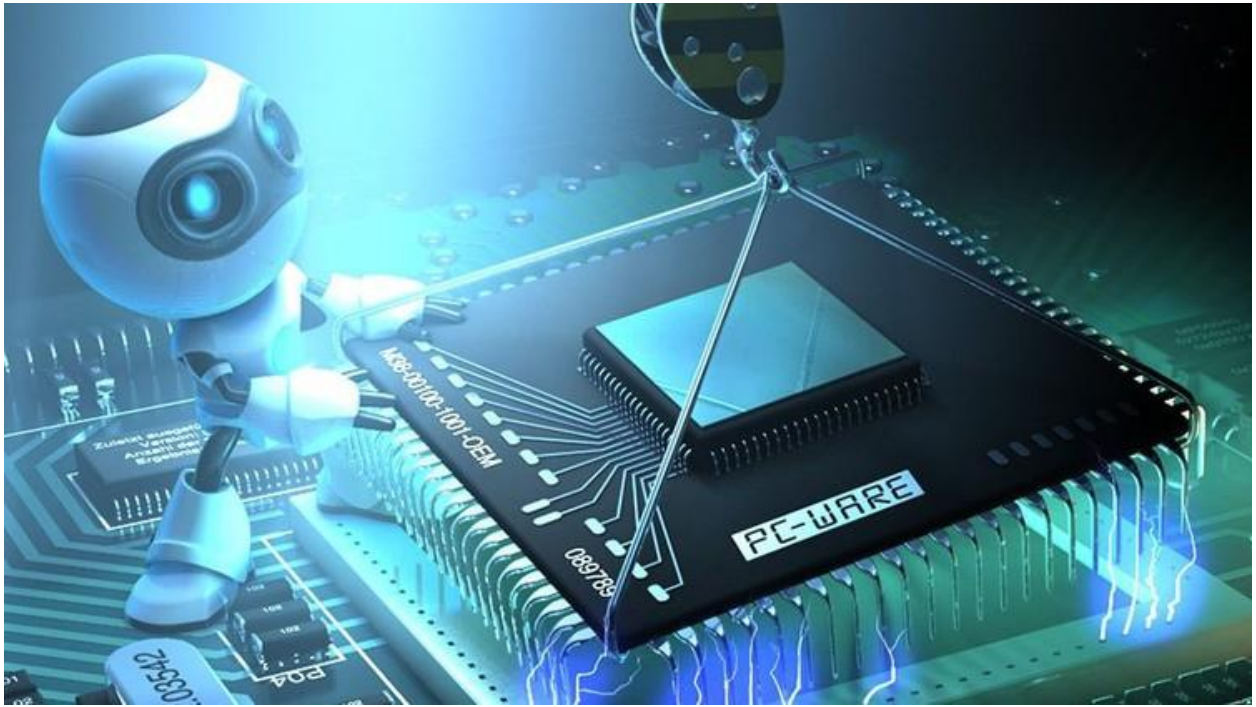
Exam 2 - Problem 2



Exam 2 - Problem 2

Given the following code written in C:

```
int exam(int a[5], int b[5], int x) {
    int c[5], y;
    y=1;
    …
}
```

Exam 2 - Problem 2

a) **Draw** the activation block of the `exam` routine, clearly indicating the offsets with respect to **`%ebp`** and the size of all fields.

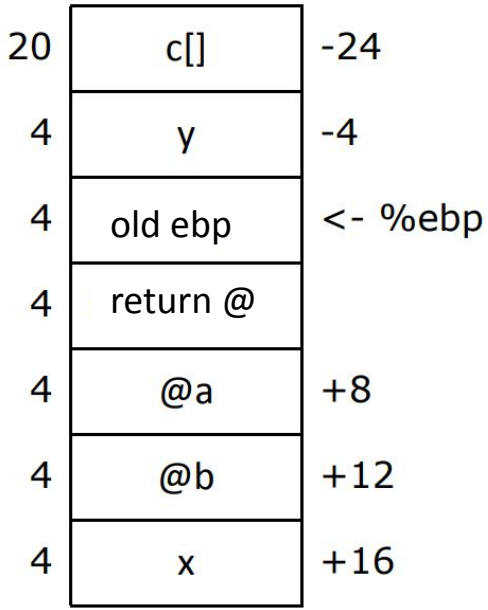
b) **Translate** the start of the `exam` routine to x86 assembler (first three lines in C up to the statement y=1;) knowing that it uses the `%eax`, `%ebx`, `%ecx` and `%edx` registers.

c) **Translate** the following C statement found in the body of the exam routine to x86 assembler.

```
y = exam(b, c, a[3]);
```

Exam 2 - Problem 2

a) **Draw** the activation block of the `exam` routine, clearly indicating the offsets with respect to **`%ebp`** and the size of all fields.



Exam 2 - Problem 2

b) **Translate** the start of the `exam` routine to x86 assembler (first three lines in C up to the statement y=1;) knowing that it uses the `%eax`, `%ebx`, `%ecx` and `%edx` registers.

```
exam:   pushl %ebp
        movl %esp, %ebp
        subl $24, %esp;
        pushl %ebx
        movl $1, -4(%ebp)
```

Exam 2 - Problem 2

c) **Translate** the following C statement found in the body of the `exam` routine to x86 assembler.

```
y = exam(b, c, a[3]);
```

```
movl 8(%ebp), %ebx      # %ebx = @a[0]
pushl 12(%ebx)          # push a[3]
leal -24(%ebp), %eax    # %eax = @c
pushl %eax              # push @c
pushl 12(%ebp)          # push @b
call exam:
addl $12, %esp
movl %eax, -4(%ebp)
```