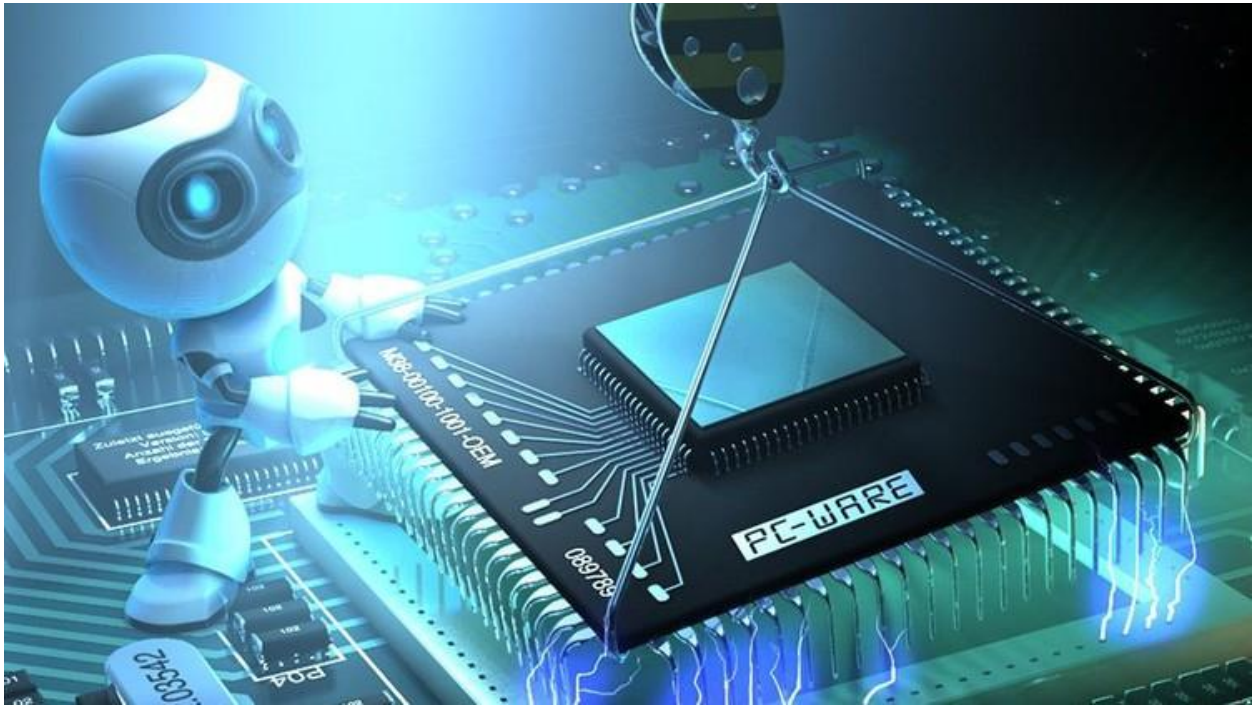
Exam 4 - Problem 2



Exam 4 - Problem 2

Given the following code written in C:

```
typedef struct {
    int id ;
    char letter;
    char sign;
    int price;
} T;

int discount(T person, int percentage) {

    if (person.id > 10) { return person.price * 2 * percentage; }
    else { return person.price * 8 * percentage; }
}
```

Lucas Bazilio - Udemy

Exam 4 - Problem 2

a) Draw how the structure `T` would be stored in memory, clearly indicating the displacements and the size of all the fields.

b) Translate the function `discount` to x86 assembler.

Exam 4 - Problem 2

a) Translate the following statement to x86 assembler, assuming it's inside the F function:

```
typedef struct {
    int id ;
    char letter;
    char sign;
    int price;
} T;
```

```
  +----------------------+
0 |          id          |   <- 4 bytes, aligned
  +----------------------+
4 |        letter        |   <- 1 byte
  +----------------------+
5 |         sign         |   <- 1 byte
  +----------------------+
6 |       (padding)      |   <- 1 byte, padding
  +----------------------+
7 |       (padding)      |   <- 1 byte, padding
  +----------------------+
8 |        price         |   <- 4 bytes, aligned
  +----------------------+
```

Exam 4 - Problem 2

b)

```
discount:
        pushl %ebp                  # save old base pointer
        movl %esp, %ebp             # set up new base pointer

        movl    8(%ebp),    %eax                    #    %eax    =    &person

        cmpl $10, 0(%eax)           # compare person.id with 10
        jle else_condition      # jump to else condition if person.id <= 10
```

Part 1/3

```
int discount(T person, int percentage) {

        if (person.id > 10) {
        return person.price * 2 * percentage;}
        else {
        return person.price * 8 * percentage; }
}
```

x86

Exam 4 - Problem 2

b)

```
# if person.id > 10
        movl 12(%ebp), %ecx     # load percentage into ecx
        movl 8(%eax), %eax       # %eax = person.price
        imull $2, %eax          # multiply person.price by 2
        imull %ecx, %eax        # multiply by percentage
        jmp end
```

x86

Part 2/3

Exam 4 - Problem 2

b)

```
else_condition:
        # if person.id <= 10
        movl 12(%ebp), %ecx     # load percentage into ecx
        movl 8(%eax), %eax       # %eax = person.price
        imull $8, %eax          # multiply person.price by 8
        imull %ecx, %eax        # multiply by percentage

end:
        movl %ebp, %esp
        popl %ebp
        ret                     # return result in eax
```

x86

Part 3/3