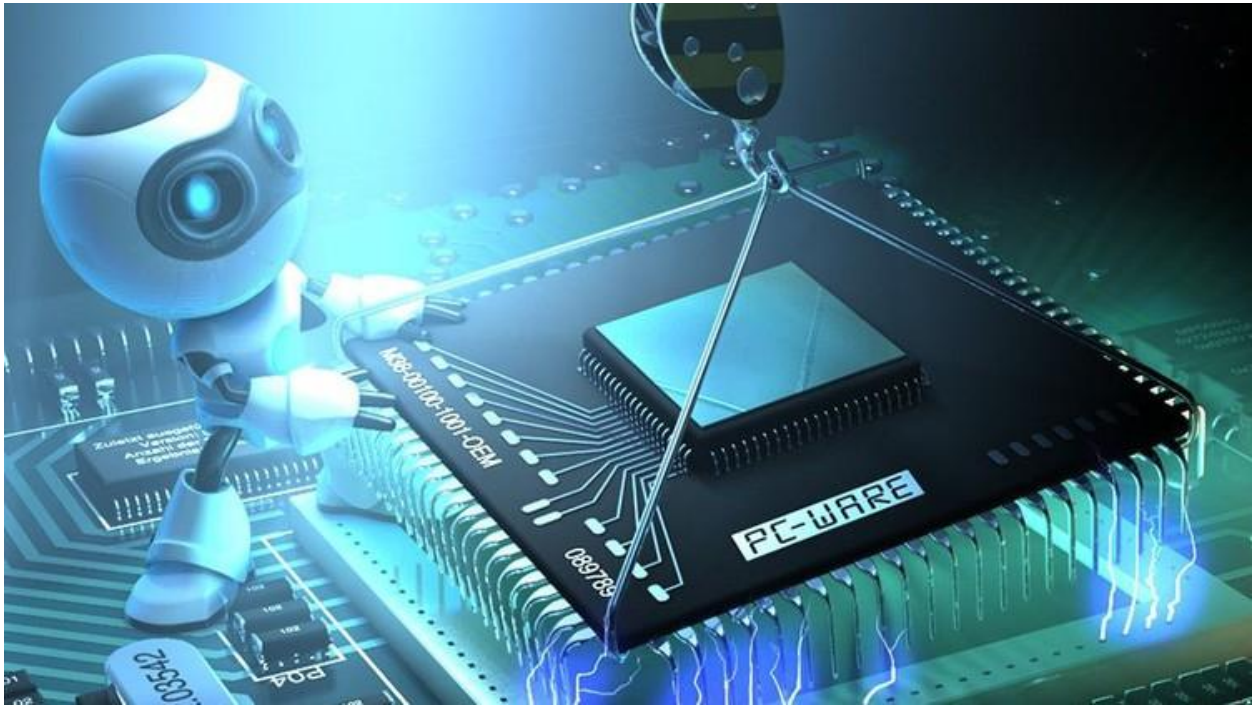
Exam 5 - Problem 2



Exam 5 - Problem 2

Given the following code written in C:

```
typedef struct {

 int number;
 char name[10];
 int floors;
} Building;

int class (Building A) {
      int found = 0;
      for (int i = 0; A.name[i] != '\0'; i++) {
            if (A.name[i] == 'a') { found = 1; break; }
      if (found == 1) {
            if (A.floors > 5) return 1;
            else return 2;
      }
      else {
            if (A.floors > 5) return 3;
            else return 4;
      }
}
```

Lucas Bazilio - Udemy

Exam 5 - Problem 2

a) Draw how the `Building` structure would be stored, clearly indicating the offsets from the start and the size of all the fields.

b) Translate the `class` subroutine to x86 assembler.

Exam 5 - Problem 2

a) Draw how the `Building` structure would be stored, indicating the offsets from the start and the size of all the fields.

```
typedef struct {

 int number;
 char name[10];
 int floors;
} Building;
```

Exam 5 - Problem 2

a) Draw how the `Building` structure would be stored, indicating the offsets from the start and the size of all the fields.

Offset	Type	Field	
0	int	number	4 bytes
4	char	name[10]	10 bytes
14	char	padding	2 bytes
16	int	floors	4 bytes

```
typedef struct {

 int number;
 char name[10];
 int floors;
} Building;
```

Total size = 20 bytes

Exam 5 - Problem 2

b)

```
class:
        pushl %ebp                      # save old base pointer
        movl %esp, %ebp                 # set up new base pointer
        movl 8(%ebp), %eax              # load address of Building A into %eax
        xorl %ecx, %ecx                 # Initialize the variable found to 0
        movl $0, %edx                   # initialize counter (i = 0)
loop:
        movzbl 4(%eax, %edx, 1), %ebx   # load current character from name into %ebx
        testb %bl, %bl                  # Check if A.name[i] is zero
        je endloop                      # If A.name[i] is zero, exit the loop
        cmpb $'a', %bl                  # Compare A.name[i] with 'a'
        jne loop                        # If A.name[i] != 'a', continue looping
        movl $1, %ecx                   # Set found to 1
        jmp endloop
```

x86

Part 1/3

Exam 5 - Problem 2

b)

```
# Check the value of found and A.floors to determine the return value
endloop:
        cmpl $1, %ecx          # Check if found is 1
        je found               # If found is 1, jump to the found label
        cmpl $5, 16(%eax)      # Compare A.floors with 5
        jle floors_le_5        # If A.floors <= 5, jump to floors_le_5
        movl $3, %eax          # Return 3
        jmp done
```

x86

Part 2/3

Exam 5 - Problem 2

b)

```
found:
        cmpl $5, 16(%eax)        # Compare A.floors with 5
        jle found_floors_le_5    # If A.floors <= 5, jump to found_floors_le_5
        movl $1, %eax            # Return 1
        jmp done
floors_le_5:
        movl $4, %eax            # Return 4 (case not found and floors <= 5)
        jmp done
found_floors_le_5:
        movl $2, %eax            # Return 2 (case not found and floors <= 5)
done:
        pop %ebp                 # Restore old base pointer
        ret                      # Return from function
```

x86

Part 3/3