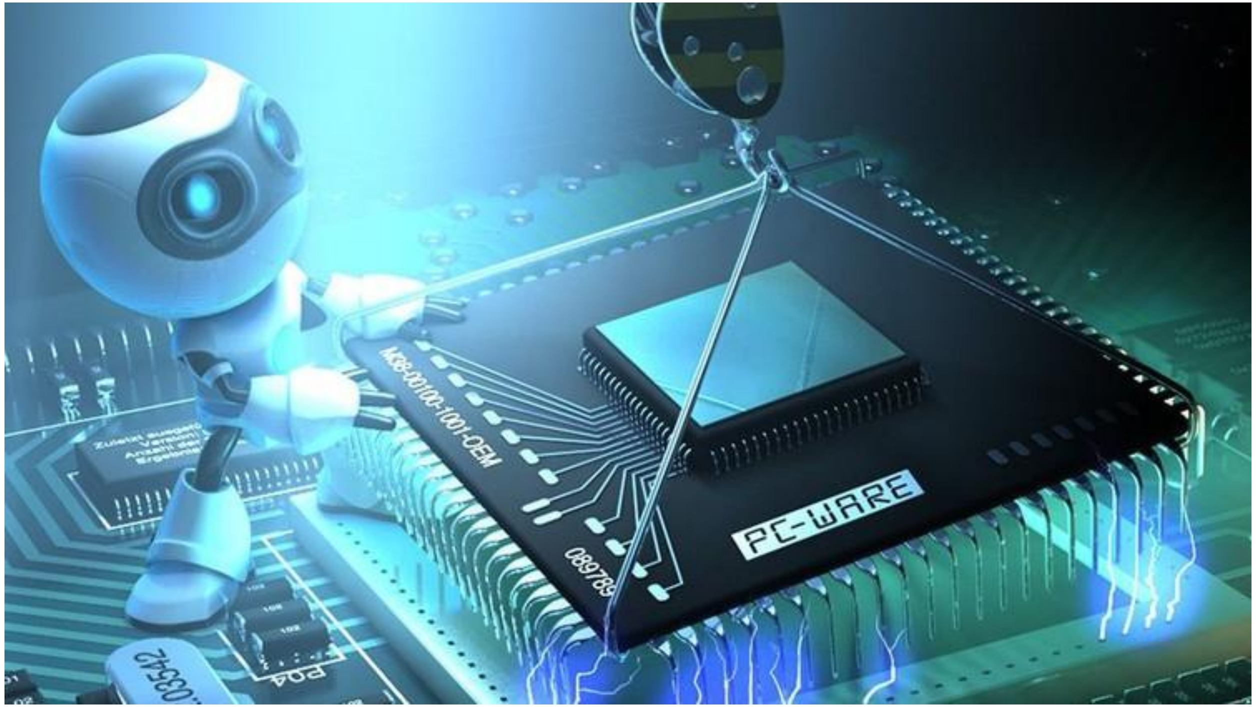
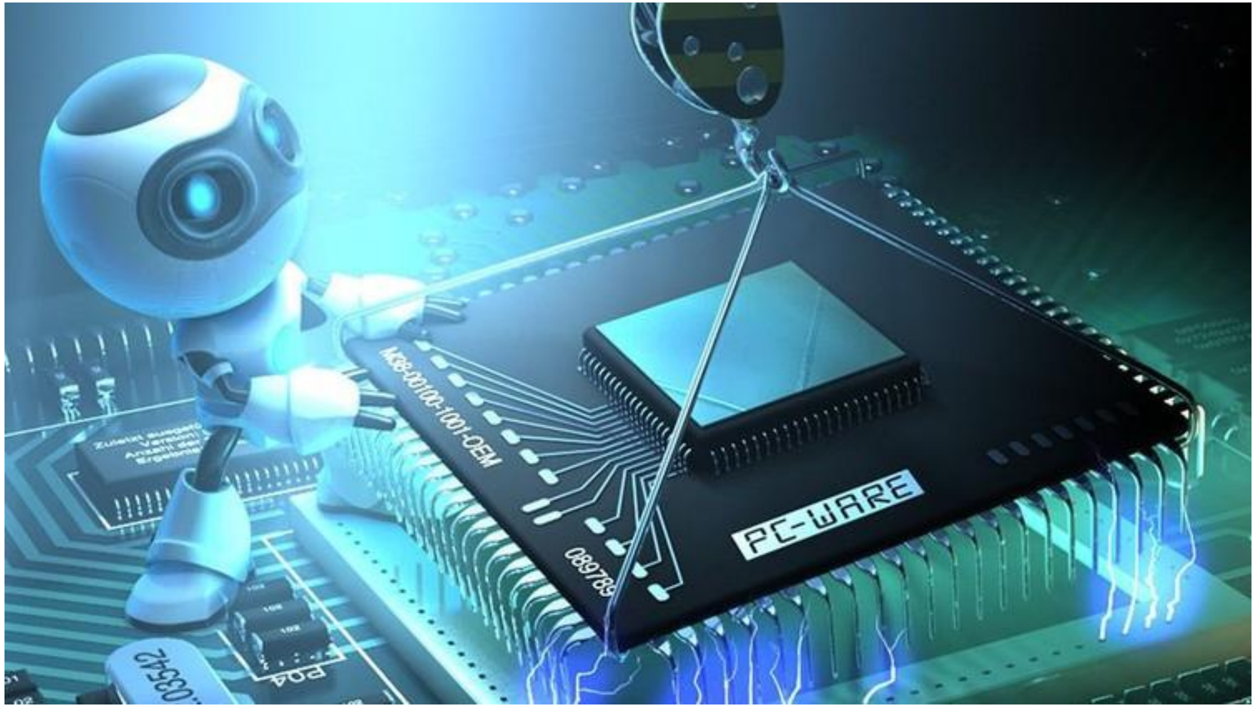
Laboratory Session 1



Practise - Problem 1



Practise - Problem 1

1. Translate this subroutine that has the following high-level code:

```
int OperationVec(int Vector[], int elements) {
// The @ of Vector is in @ 8[ebp] and the
// value of the variable elements in @ 12[ebp]
int i;          // i is in @ -8[ebp]
int res;     // res is in @ -4[ebp]
res=Vector[0];

for (i=1;i<elements;i++)
  if (Vector[i]<res)
    res=Vector[i];

return res;
}
```

Lucas Bazilio - Udemy

Practise - Problem 1

1.

Part 1/3

```
.text
        .align 4
        .globl OperationVec
        .type OperationVec, @function
OperationVec:
        pushl       %ebp
        movl %esp, %ebp
        subl  $16, %esp
        pushl       %ebx
        pushl       %esi
        pushl       %edi
        movl 8(%ebp), %eax          # %eax ← @Vector ≡ @Vector[0]
        movl (%eax), %eax           # %eax ← Vector[0]
        movl %eax, -4(%ebp)         # res ← Vector[0]
```

x86

Lucas Bazilio - Udemy

Practise - Problem 1

1.

Part 2/3

```
        movl $1, %ecx                            # ecx = 1 (= i)

for:
        cmpl 12(%ebp), %ecx
        jge         endfor

        movl 8(%ebp, %ecx, 4), %edx              #Vector[i]
        cmpl -4(%ebp), %edx
        jge endif
        movl %edx, -4(%ebp)                      #res = Vector[i];
```

x86

Lucas Bazilio - Udemy

Practise - Problem 1

1.

Part 3/3

```
endif:
        incl %ecx
        jmp for
endfor:
        movl %ecx, -8(%ebp)          # i = ecx
        movl -4(%ebp), %eax          # %eax ← res
        popl %edi
        popl %esi
        popl %ebx
        movl %ebp,%esp
        popl %ebp
        ret
```

x86

Lucas Bazilio - Udemy