

JOSÉ LUIS CUMBRERA SÁNCHEZ
TEODORO MARTÍNEZ MÁRQUEZ

PL PRACTICE

Structure:

```
int main(){  
    variable declarations;  
  
    assignments;  
    printf instruction;  
    if, if-else, while sentences;  
  
    return 0;  
}
```

Variables.

We accept integers, vectors and pointers.

```
int main(){  
    int a,b,c;  
  
    int v[10];  
  
    int ***p;  
  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $56, %esp  
  
    movl $0, %eax  
  
    movl %ebp, %esp  
    popl %ebp  
    ret
```

Internal definitions. (uncomment Nodo.Escribir)

```
int main(){  
    int a,*b,***c,*d[3],e[4][2];  
  
    return 0;  
}
```

```
Var  a :  
Entero.  
Var  b :  
Puntero a  
Entero.  
Var  c :  
Puntero a  
Puntero a  
Puntero a  
Entero.  
Var  d :  
NodoArray de 3  
Puntero a  
Entero.  
Var  e :  
NodoArray de 4  
NodoArray de 2  
Entero.  
  
{'a': <__main__.NodoInt at 0x29ca16f5908>,  
'b': <__main__.NodoPuntero at 0x29ca16f5278>,  
'c': <__main__.NodoPuntero at 0x29ca16f54a8>,  
'd': <__main__.NodoArray at 0x29ca16f58d0>,  
'e': <__main__.NodoArray at 0x29ca16f5860>}
```

Assignment

```
int main(){  
    int a;  
  
    a=2*3+8/19+1-2;  
  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $4, %esp  
  
    movl $2, %eax  
    pushl %eax  
  
    movl $3, %eax  
    movl %eax, %ebx  
    popl %eax  
    imull %ebx, %eax  
  
    pushl %eax  
  
    movl $8, %eax  
    pushl %eax  
  
    movl $19, %eax  
  
    movl %eax, %ebx  
    popl %eax  
    cdq  
    idivl %ebx
```

```
    movl %eax, %ebx  
    popl %eax  
    addl %ebx, %eax  
  
    pushl %eax  
  
    movl $1, %eax  
  
    movl %eax, %ebx  
    popl %eax  
    addl %ebx, %eax  
  
    pushl %eax  
  
    movl $2, %eax  
  
    movl %eax, %ebx  
    popl %eax  
    subl %ebx, %eax  
  
    movl %eax, -4(%ebp)  
  
    movl $0, %eax  
  
    movl %ebp, %esp  
    popl %ebp  
    ret
```

If sentence.

```
int main(){  
    int a;  
  
    a=1;  
  
    if(a==1){  
        a=2;  
    }  
  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $4, %esp  
  
    movl $1, %eax  
  
    movl %eax, -4(%ebp)  
  
    movl $1, %eax  
  
    movl -4(%ebp), %eax  
    cmpl $1, %eax  
    jne final0  
  
    movl $2, %eax  
  
    movl %eax, -4(%ebp)  
  
final0:  
  
    movl $0, %eax  
  
    movl %ebp, %esp  
    popl %ebp  
    ret
```

while sentence.

```
int main(){  
    int a,b,c;  
  
    a=12;  
  
    while(a<=12){  
        a=a-1;  
    }  
  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $12, %esp  
  
    movl $12, %eax  
  
    movl %eax, -4(%ebp)  
  
start0:  
    movl $12, %eax  
  
    movl -4(%ebp), %eax  
    cmpl $12, %eax  
    jg final0  
  
    pushl %eax  
  
    movl $1, %eax  
  
    movl %eax, %ebx  
    popl %eax  
    subl %ebx, %eax  
  
    movl %eax, -4(%ebp)  
  
    jmp start0  
..
```

```
final0:  
  
    movl $0, %eax  
  
    movl %ebp, %esp  
    popl %ebp  
    ret
```

If-else sentence.

```
int main(){  
    int a,b,c;  
    a=1;  
    if(a==1){  
        b=2;  
    }else{  
        b=3;  
    }  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $12, %esp  
  
    movl $1, %eax  
  
    movl %eax, -4(%ebp)  
  
    movl $1, %eax  
  
    movl -4(%ebp), %eax  
    cmpl $1, %eax  
    jne final0  
  
    movl $2, %eax  
  
    movl %eax, -8(%ebp)  
  
    jmp final1  
final0:  
    movl $3, %eax  
  
    movl %eax, -8(%ebp)  
  
final1:  
  
    movl $0, %eax
```

```
movl %ebp, %esp  
popl %ebp  
ret
```

Printf.

```
int main(){  
    int a,b,c;  
    a=1;  
    b=2;  
    c=3;  
    printf("%d %d %d",a,b,c);  
    return 0;  
}
```

```
.text                                popl %ebp  
.globl main                          ret  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $12, %esp  
  
    movl $1, %eax  
    movl %eax, -4(%ebp)  
  
    movl $2, %eax  
    movl %eax, -8(%ebp)  
  
    movl $3, %eax  
    movl %eax, -12(%ebp)  
  
    pushl -4(%ebp)  
    pushl -8(%ebp)  
    pushl -12(%ebp)  
    puhs1 "%d%d%d"  
    call printf  
  
    addl $16, %esp  
    movl $0, %eax  
  
    movl %ebp, %esp
```

If with &&.

```
int main(){
    int a,b,c;

    a=1;

    b=3;

    if(a==1 && b>=3){
        c=7;
    }

    return 0;
}
```

```
.text
.globl main
.type main, @function
main:
    push %ebp
    movl %esp, %ebp
    subl $12, %esp

    movl $1, %eax
    movl %eax, -4(%ebp)

    movl $3, %eax
    movl %eax, -8(%ebp)

    movl $1, %eax
    movl -4(%ebp), %eax
    cmpl $1, %eax
    jne final0

    movl $3, %eax
    movl -8(%ebp), %eax
    cmpl $3, %eax
    jl final0

    movl $7, %eax

    movl %eax, -12(%ebp)

final0:
    movl $0, %eax
    movl %ebp, %esp
    popl %ebp
    ret
```

If with ||.

```
int main(){  
    int a,b,c;  
  
    a=1;  
  
    b=3;  
  
    if(a==1 || b>=3){  
        c=7;  
    }  
  
    return 0;  
}
```

```
.text  
.globl main  
.type main, @function  
main:  
  
    push %ebp  
    movl %esp, %ebp  
    subl $12, %esp  
  
    movl $1, %eax  
  
    movl %eax, -4(%ebp)  
  
    movl $3, %eax  
  
    movl %eax, -8(%ebp)  
  
    movl $1, %eax  
  
    movl -4(%ebp), %eax  
    cmpl $1, %eax  
    jne final0  
  
    movl $3, %eax  
  
    movl -8(%ebp), %eax  
    cmpl $3, %eax  
    jge final0  
    jmp notOr0  
  
final0:
```

```
    movl $7, %eax  
  
    movl %eax, -12(%ebp)  
  
notOr0:  
  
    movl $0, %eax  
  
    movl %ebp, %esp  
    popl %ebp  
    ret
```

If inside an if.

```
int main(){
    int a,b,c;

    a=1;

    if(a<=1){
        if(a<=2){
            b=14;
        }
    }

    return 0;
}
```

```
.text
.globl main
.type main, @function
main:

    push %ebp
    movl %esp, %ebp
    subl $12, %esp

    movl $1, %eax

    movl %eax, -4(%ebp)

    movl $1, %eax

    movl -4(%ebp), %eax
    cmpl $1, %eax
    jg final0

    movl $2, %eax

    movl -4(%ebp), %eax
    cmpl $2, %eax
    jg final0

    movl $14, %eax

    movl %eax, -8(%ebp)

final0:
```

```
final1:

    movl $0, %eax

    movl %ebp, %esp
    popl %ebp
    ret
```