

PL PRACTICE

JOSÉ LUIS CUMBRERA SÁNCHEZ
TEODORO MARTÍNEZ MÁRQUEZ

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

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
    movbl %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:
    movl $0, %eax
    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:
    movl %ebp, %esp
    popl %ebp
    ret

    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
```

Printf.

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

```
.text
.globl main
.type main, @function
main:
    pushl %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)
    puhsq "%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           movl %eax, -12(%ebp)
.type main, @function
main:                final0:
                    push %ebp
                    movl %esp, %ebp      movl $0, %eax
                    subl $12, %esp
                    movl $1, %eax       movl %ebp, %esp
                    movl %eax, -4(%ebp)  popl %ebp
                                ret
                    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
```

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:
    movl $7, %eax
    movl %eax, -12(%ebp)
    push %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl $1, %eax
    movl %eax, -4(%ebp)
    movl $0, %eax
    movl %ebp, %esp
    popl %ebp
    ret
notOr0:
    movl $0, %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:
```

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:                                final1:
    movl $0, %eax
    push %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl %ebp, %esp
    popl %ebp
    ret
    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:
```