

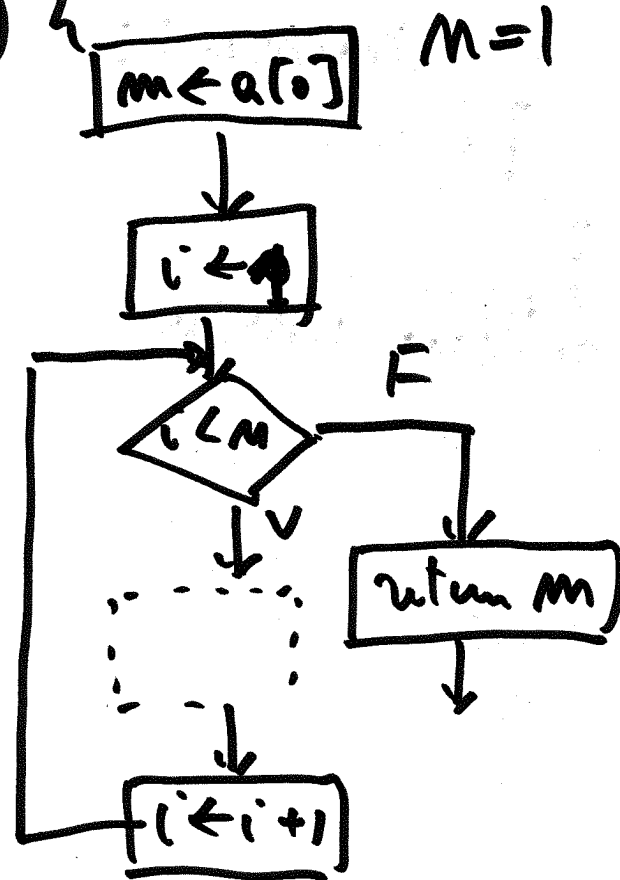
7	-1	2	5
---	----	---	---

2050	3721	1820
------	------	------

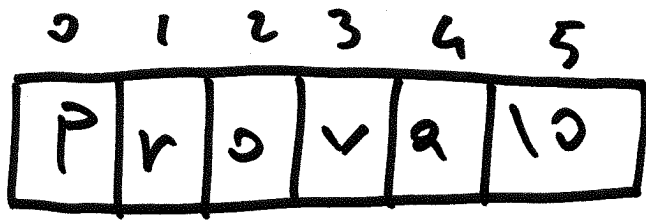
```

int minimo (int a[], int n) ← Open annotation
{
  int m = 10000, a[0]; ← Solo se m > 0
  int i;
  for (i = 0; i < n; i++) {
    if (a[i] < m) {
      m = a[i];
    }
  }
  return m;
}

```

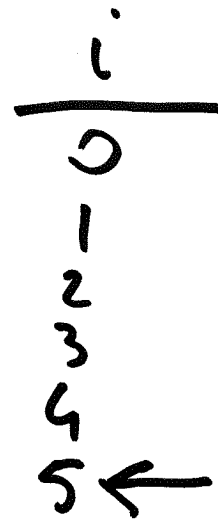
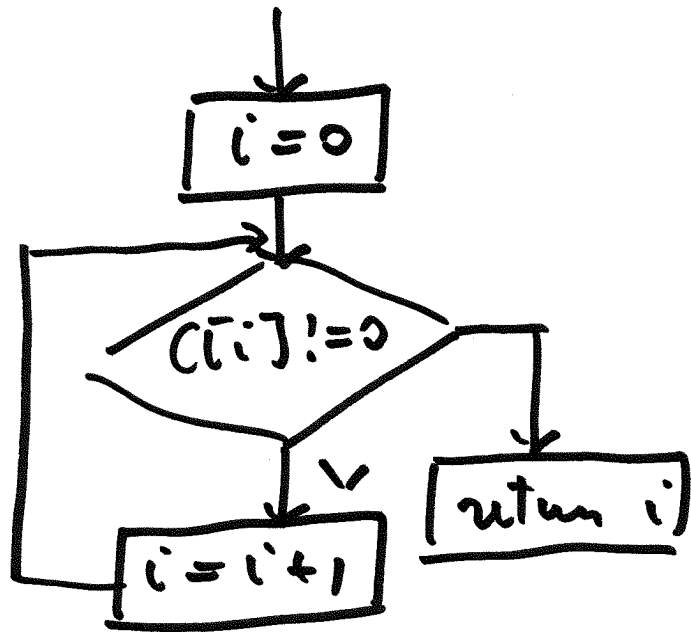


```
int findMinimum (int a[], int n)
{
    int iMin = 0;
    int i;
    for (i = 1; i < n; i++) {
        if (a[i] < a[iMin]) {
            iMin = i;
        }
    }
    return iMin;
}
```



char c[]

```
for (i=0; c[i] != 0; i++);
```



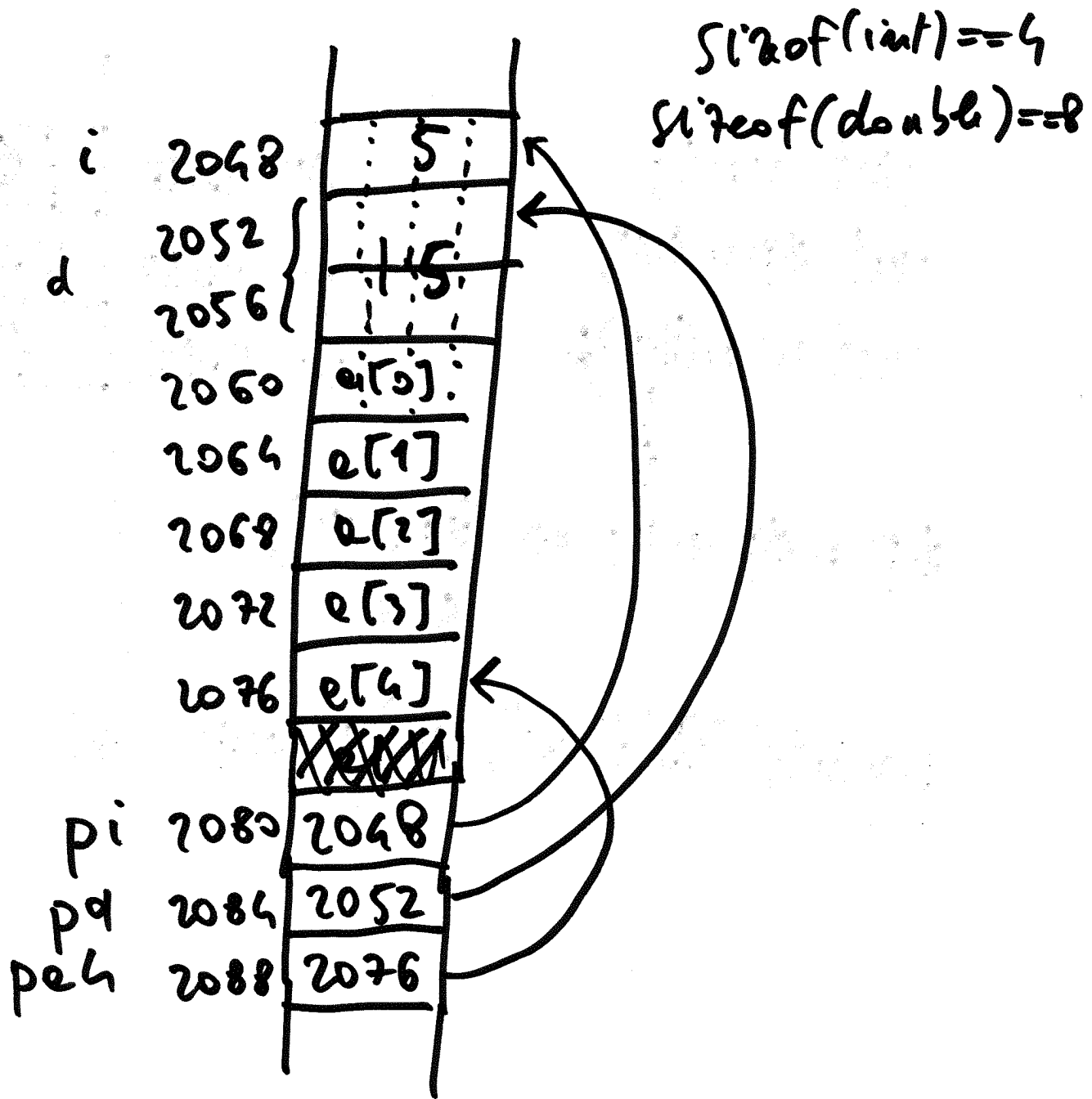
```
for (i=0; c[i]; i++);
```

$c[i] != 0$

```

int i = 5;
double d = 1.5;
int a[5];
int *pi = &i;
double *pd = &d;
int *pa[4] = &a[4];

```



int v = 5;

int *p = &v;

int w = (*p);

5

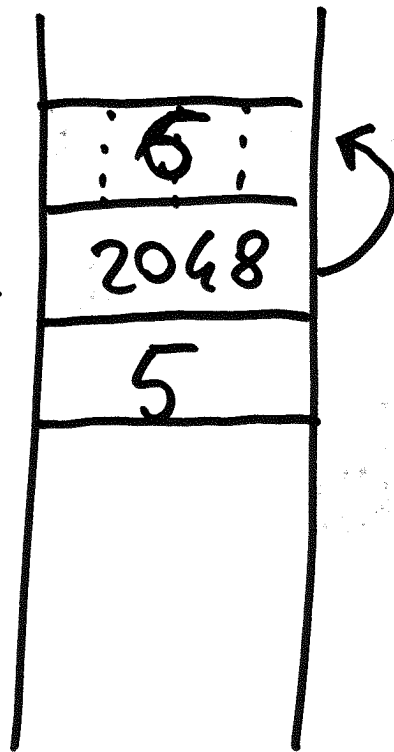
*p = *p + 1; \Rightarrow (*p) + 1;
5 + 1

printf("%d\n", v); \Rightarrow 6

v 2048

p 2052

w 2056



define N 10

int a[N];

a[5]

a he tipo "puntador a int"
int *

⇓

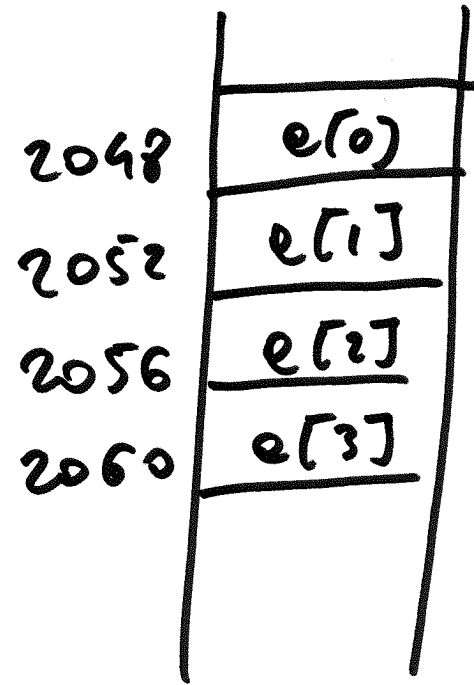
&a[0]

int k=3;

$$(\&a + k) = 2048 + 3 = 2051$$

$$= \underline{2048 + k * \text{sizeof}(int)}$$

$$= 2048 + 3 \times 4 = 2048 + 12 = 2060$$



$$\text{Mem}(a[k]) = \text{Mem}(a[0]) + k * \text{sizeof}(int)$$

se e è un array di elm. di tipo T

e k è un intero

allora

$(e+k)$ è l'indirizzo di memoria e pointer del
quale è memorizzato $e[k]$

$$\boxed{(e+k) \equiv \& e[k]}$$

$$e+k = e + k * \text{sizeof}(T)$$

```
int e[] = {15, 16, 12};
```

```
int *p, *q;
```

```
p = a;
```

```
*p = -1;
```

```
q = p + 1;
```

```
q++;
```

```
*q = -3;
```

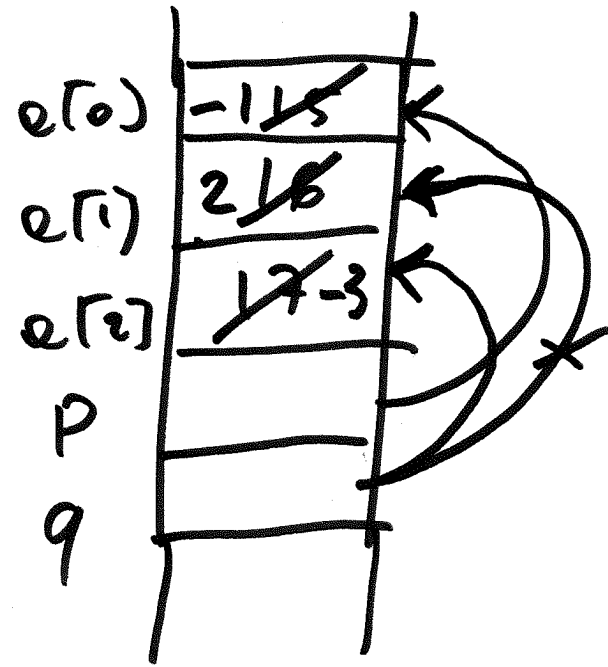
```
*(p + 1) = 2
```

$p + \text{sizeof}(\text{int})$

$p + 1 \equiv \&(e[1])$

$q = q + 1;$

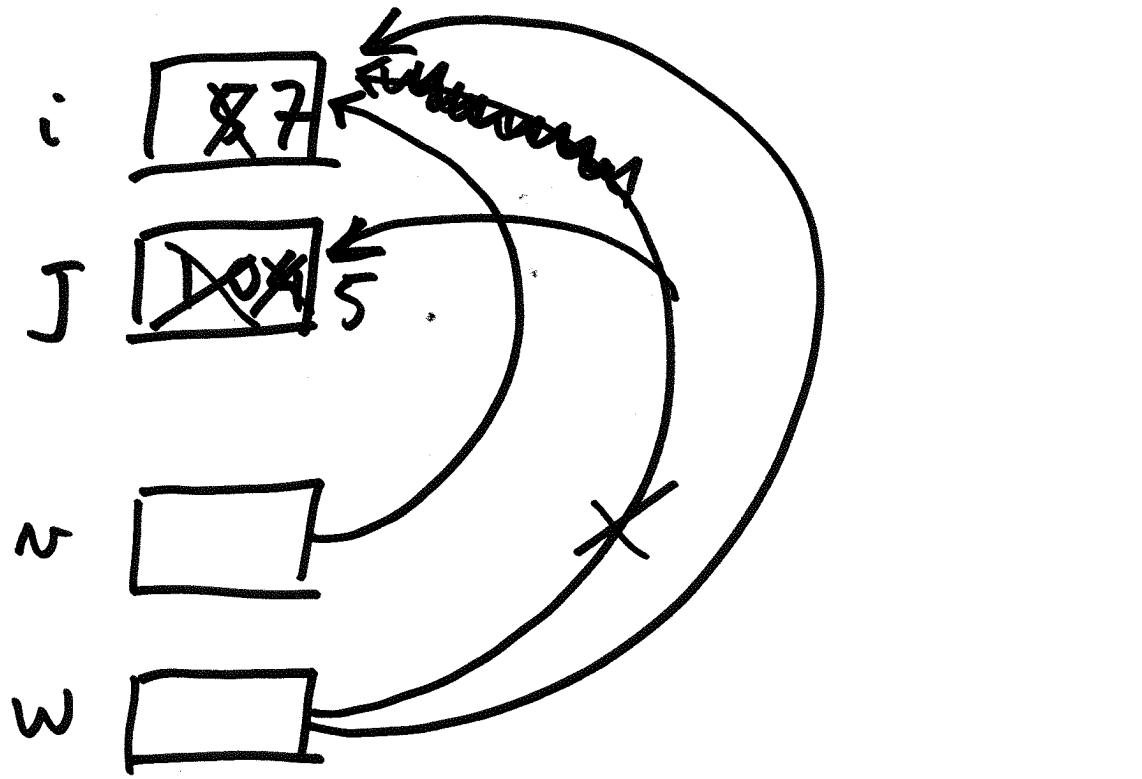
$e[] = \{ -1, 2, -3 \}$




```

int i = 5, j = 10;
int *v = &i;
int *w;
w = &j;
*v = 4;
*w = *v; 5
w = v;
*w = 7;
printf("%d %d %d %d\n",
       i, j, *v, *w)

```



7 5 7 7

```
int e[] = {11, 7};  
int *w = &e[1];  
int i = *w;  
*w = 0;
```

```
printf("%d %d\n",  
       e[0], e[1]); => 11 0
```

```
int *ptr_e = NULL; ptr_e  
X printf("%d\n", *ptr_e);
```

