Conversione in binario

x in base 10 ⟶ base 2

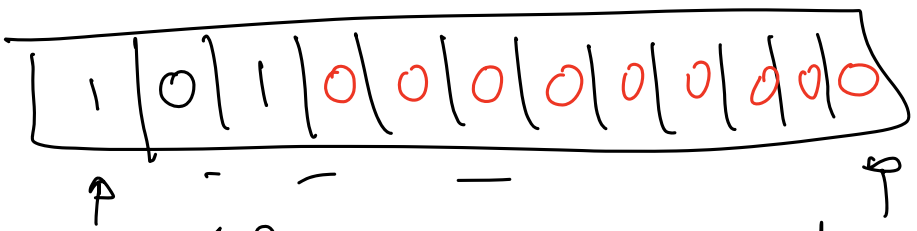
x = 17 | 17 / 2 | 1 | Cifra 0
8 / 2 | 0 | Cifra 1

```
#define NBIT 16

int main() {
    int x;
    printf("inserisci x < %d : ", (int)pow(2, NBIT));
    scanf("%d", &x);

    int binario[NBIT] = {0};
    int resto;
    int cifra=0;
    do {
        resto = x % 2;
        binario[cifra] = resto;
        x /= 2;
        cifra++;
    } while( x != 0 );
```

x = 5

cifra	resto
0	1
1	0
2	1

binario[NBIT]



binario[0] ... binario[NBIT-1]

```
    printf("%d in base 10 -> ", x);
    for(int i=0; i<NBIT; i++) {
        printf("%d", binario[i]);
    }
}
```

printf("\n");

7 in base 10: 10100000000000000

```
for (int i= NBIT-1; i>=0; i--) {
    printf("%d", binario[i]);
}
```

7 in base 10 ⟶ 00000000000000101

Base b > 10

```
char Conv[NBIT];
```

≡

```
Conv[11] = 'B'
```

```
int dati[7];
```

[0]	[1]	[2]	[3]	[4]	[5]	[6]
2	-2	3	4	-1	0	9

```
int studenti[160000];
```

2 problemi comuni:

- ordinamento
- ricerca

Algoritmo BubbleSort

dati[N]

[0]	[1]	[2]	[3]	[4]	[5]	[6]
2	-2	3	4	-1	0	9

N: lunghezza array

int i = 0;

int j = i+1 - - - - - N-1

i = 0 j = N-1 dati[j] > dati[j-1]

i = 0 j = N-2 j-1 = N-3

i = 0 j = N-3 j-1 = N-4 dati[j] < dati[j-1].

Scambio 4 ⟷ -1

temp = dati[j] -1

dati[j] = dati[j-1] 4 al posto di -1

dati[j-1] = temp 4 sovrascritto da -1

2	-2	3	-1	4	0	9

i = 0 j = N-4 j-1 = N-5

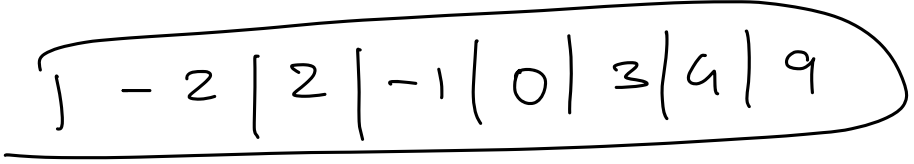
2	-2	-1	3	4	0	9

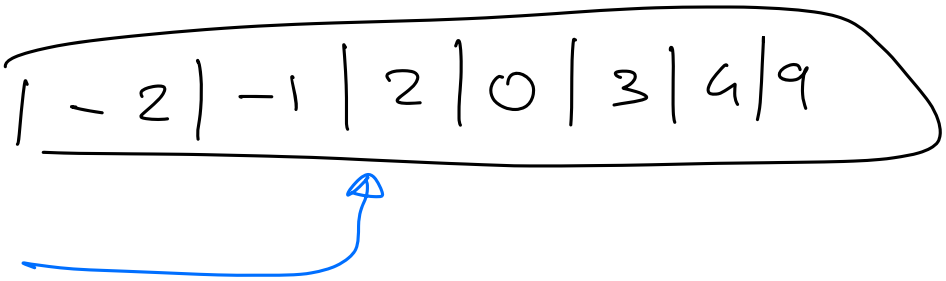
i = 0 j = N-5 j-1 = N-6

-2	2	-1	3	4	0	9

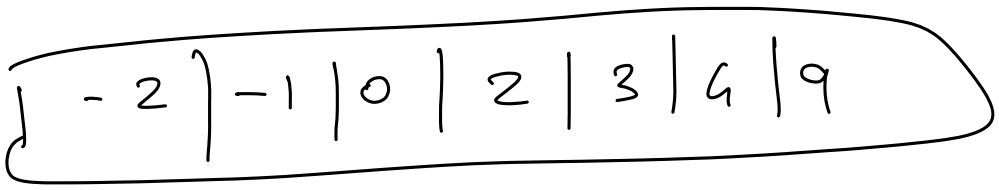
i = 1

-2	2	-1	3	0	4	9



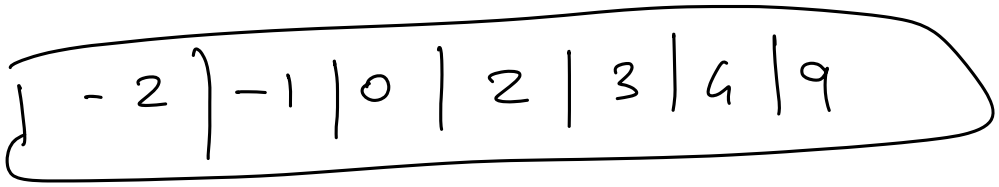


i=2




```
int dati[N];
int i, j;

for(i=0; i<N; i++) {
    for(j=N-1; j>i; j--) {
        if(dati[j] < dati[j-1]) {
            temp = dati[j];
            dati[j] = dati[j-1];
            dati[j-1] = temp;
        }
    } // ciclo j
}// ciclo i
```

Ricerca Binaria (su array ordinati)



$x = 5$ Contenuto nell'array ?

```
int inizio = 0, fine = N;
int mezzo = (N-1)/2;
if (x < dati[mezzo]) {
    fine = mezzo
} else {
    inizio = mezzo;
}
```