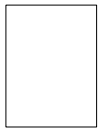
Esempio

INPUT: 0 3 4 5 .... EOF



2^0

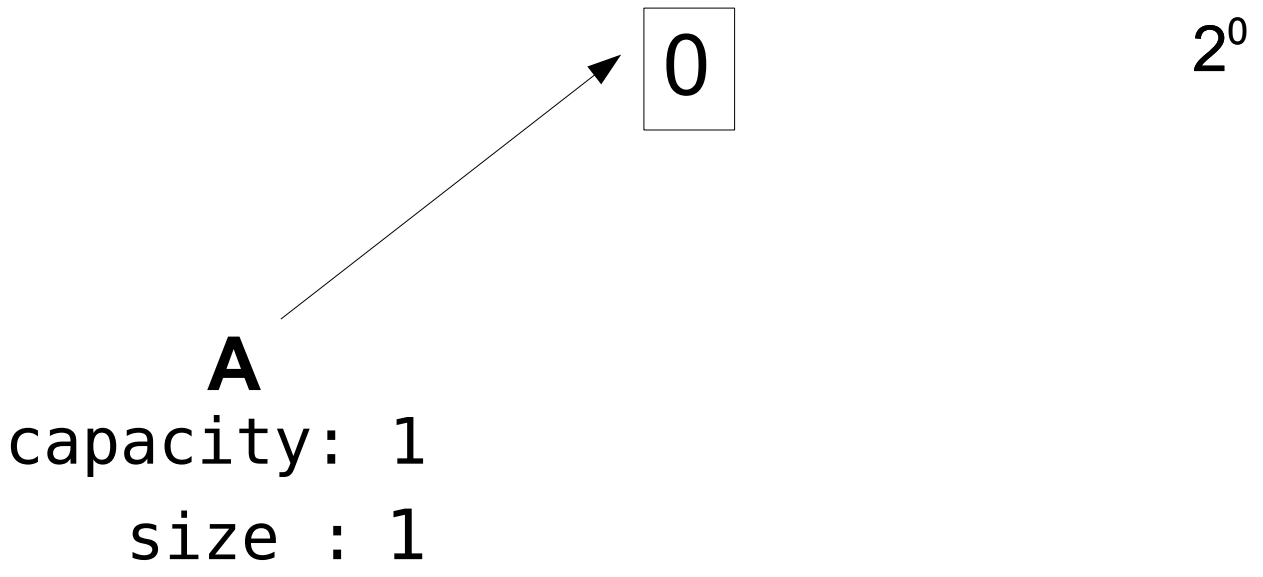
A

capacity: 1

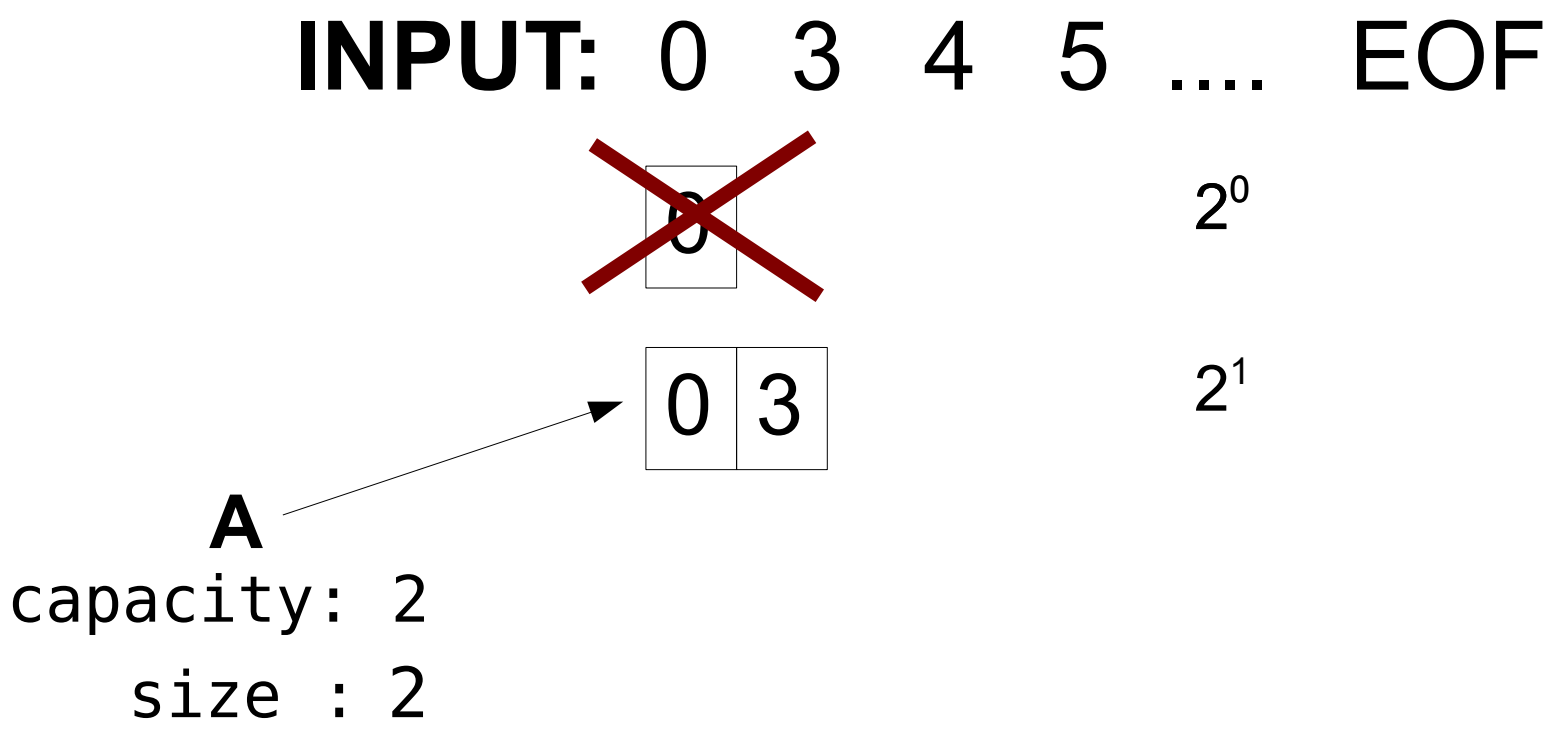
size : 0

Esempio

INPUT: 0 3 4 5 .... EOF

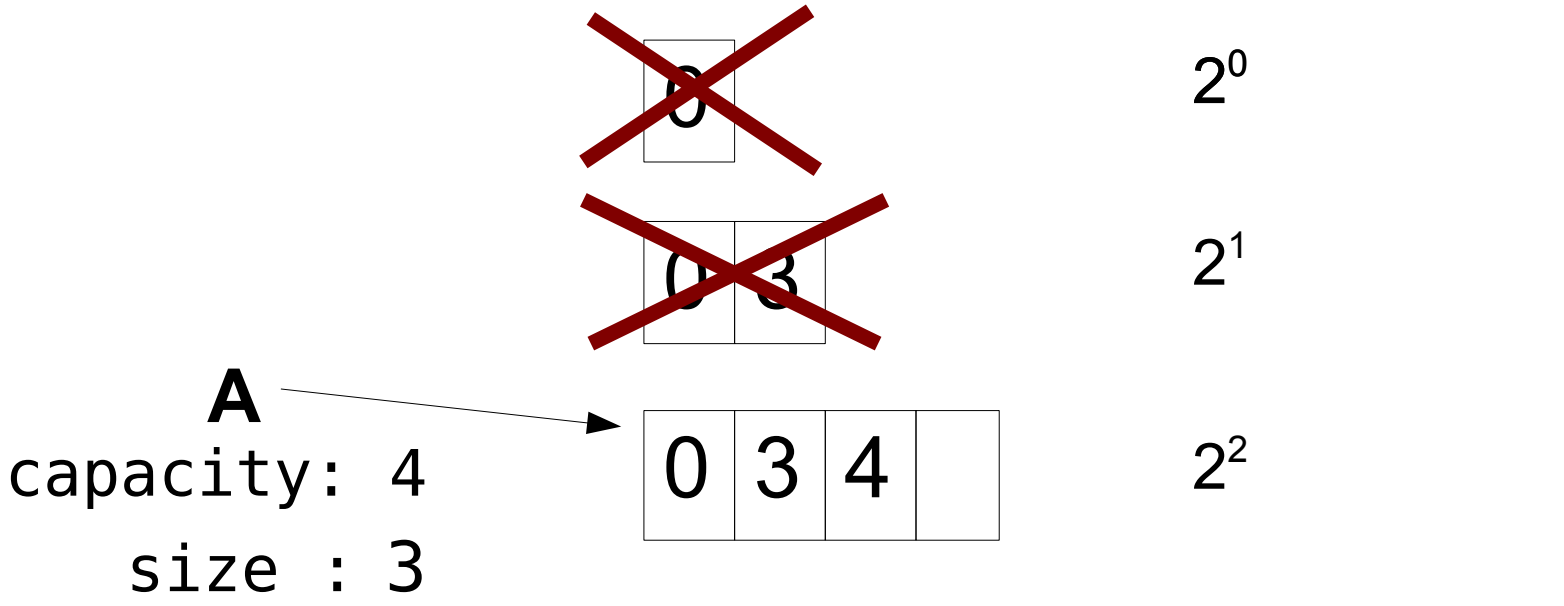


Esempio



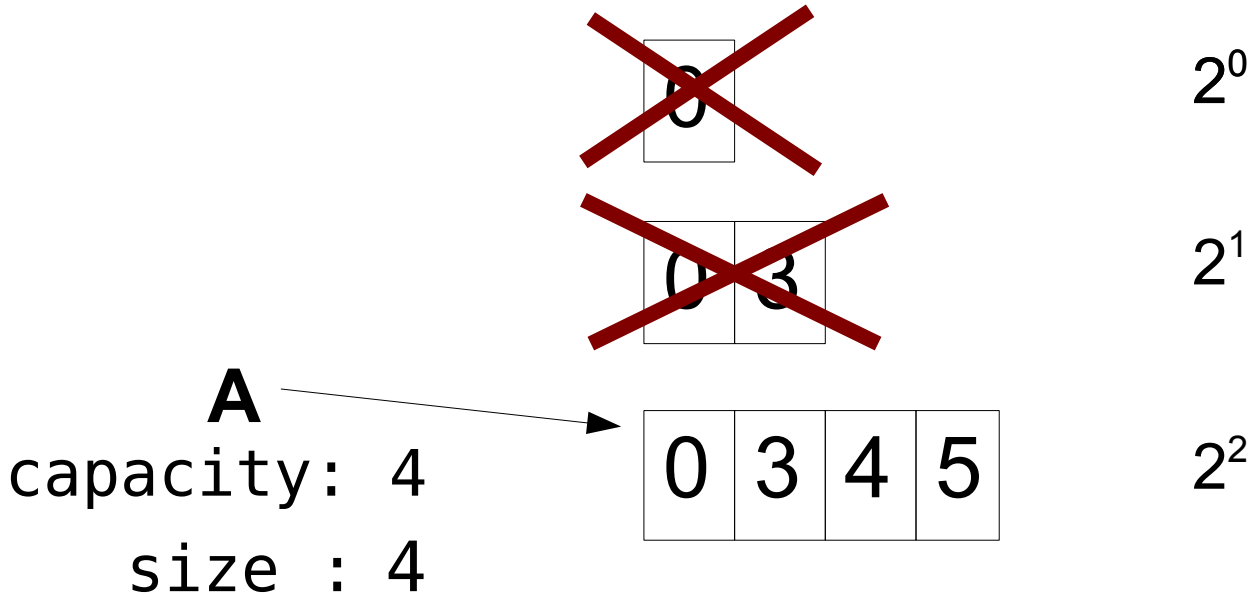
Esempio

INPUT: 0 3 4 5 .... EOF



Esempio

INPUT: 0 3 4 5 .... EOF



```
int *A, *tmp, size=0, capacity=1, e;

while (scanf("%d",&e) == 1){

  if ( size == capacity ){

    tmp = malloc((2*capacity)*sizeof(int));

    for (i=0; i<size; i++)
        tmp[i]=A[i];

    capacity = 2*capacity;
    free(A);
    A = tmp;
  }

  A[size++] = e;
}
```

```
int *A, *tmp, size=0, capacity=1, e;

while (scanf("%d",&e) == 1){

  if ( size == capacity ){

    tmp = malloc((2*capacity)*sizeof(int));

    for (i=0; i<size; i++)
        tmp[i]=A[i];

    capacity = 2*capacity;
    free(A);
    A = tmp;
  }

  A[size++] = e;
}
```

```
int *A, *tmp, size=0, capacity=1, e;

while (scanf("%d",&e) == 1){

  if ( size == capacity ){

    tmp = malloc((2*capacity)*sizeof(int));

    for (i=0; i<size; i++)
        tmp[i]=A[i];

    capacity = 2*capacity;
    free(A);
    A = tmp;
  }

  A[size++] = e;
}
```

```
int *A, *tmp, size=0, capacity=1, e;

while (scanf("%d",&e) == 1){

  if ( size == capacity ){

    tmp = malloc((2*capacity)*sizeof(int));

    for (i=0; i<size; i++)
        tmp[i]=A[i];

    capacity = 2*capacity;
    free(A);
    A = tmp;
  }

  A[size++] = e;
}
```

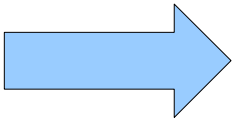
```
int *A, *tmp, size=0, capacity=1, e;

while (scanf("%d",&e) == 1){

  if ( size == capacity ){

    tmp = malloc((2*capacity)*sizeof(int));

    for (i=0; i<size; i++)  //tmp = A !?!
        tmp[i]=A[i];

    capacity = 2*capacity;
    free(A);
    A = tmp;
    //free(A)... Doh!
  }
  A[size++] = e;
}
```

Esercizio (per casa):

L'utente inserisce una sequenza di stringhe (terminate da \n), senza specificare quante. Scrivere un programma che legga la sequenza e, *mediante doubling*, inserisca in un array di char i caratteri della concatenazione di tutte le stringhe

Esempio:

Do
Re
Mi
Fa
Sol
...



Array:

D o R e M i F a S o l ....

Esercizio (per casa):

L'utente inserisce una sequenza di stringhe (terminate da \n), senza specificare quante. Scrivere un programma che legga la sequenza e, mediante doubling, inserisca in un array di char i caratteri della concatenazione di tutte le stringhe

Note importanti:

-Per inviare l'EOF da tastiera usare Ctrl+D (Ctrl+Z in Win)

-Assumere le stringhe di lunghezza massima 100

-*Tutti i caratteri di ogni stringa vanno inseriti insieme:*
la capacità del blocco dev'essere ridimensionata alla più piccola potenza del due maggiore di size.

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

2 4 8 5 7 6

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

2 4 8 5 7 6

2 4 8 5 7 6

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

2 4 8 5 7 6

2 4 8 5 7 6

2 4 5 8 7 6

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

2 4 8 5 7 6

2 4 8 5 7 6

2 4 5 8 7 6

2 4 5 6 7 8

Selection Sort

Uno dei più semplici algoritmi di sorting:

- sistema gli elementi in ordine dal più piccolo

8 4 2 5 7 6

2 4 8 5 7 6

2 4 8 5 7 6

2 4 5 8 7 6

2 4 5 6 7 8

2 4 5 6 7 8

Pseudo-Codice

```
int i,j,min,tmp;
for (i=0; i<n; i++) {

  min =i;                  /* calcola min di A[j,n]*/


  for(j=i; j < n; j++){
    if (A[j] < A[min])
            min = j;
  }

   tmp = A[i];          /* scambia A[min] e A[i]*/
   A[i] = A[min];
   A[min] = tmp;
}
```

Pseudo-Codice

```
int i,j,min,tmp;
for (i=0; i<n; i++) {

  min =i;                    /* calcola min di A[j,n]*/


  for(j=i; j < n; j++){
    if (A[j] < A[min])
            min = j;
  }

   tmp = A[i];           /* scambia A[min] e A[i]*/
   A[i] = A[min];
   A[min] = tmp;
}
```

Pseudo-Codice

```
int i,j,min,tmp;
for (i=0; i<n; i++) {

  min =i;              /* calcola min di A[j,n]*/

  for(j=i; j < n; j++){
    if (A[j] < A[min])
            min = j;
  }

   tmp = A[i];         /* scambia A[min] e A[i]*/
   A[i] = A[min];
   A[min] = tmp;
}
```

Esercizio: adattare il codice del selection sort affinchè ordini un array di stringhe. Scrivere la funzione:

```
void StringSort(char **as, int len);
```

- **as:** array di char *, **len:** lunghezza di as

che riordina gli elementi di **as** in modo che *le stringhe da essi puntate siano in ordine lessicografico*.

Note. Dichiarate l'input all'interno del programma. Es.:

```
char* as[4] = {“one”, “two”,
               “three”,“four”};
```

Esercizio (per casa)

Considerare la funzione StringSort di prima.

Leggere da terminale un intero *N* e una sequenza di *N* stringhe (di lunghezza massima 100). Utilizzare quindi la funzione StringSort per ordinare la sequenza di stringhe così ottenute.