CMSC 216
Introduction to Computer Systems

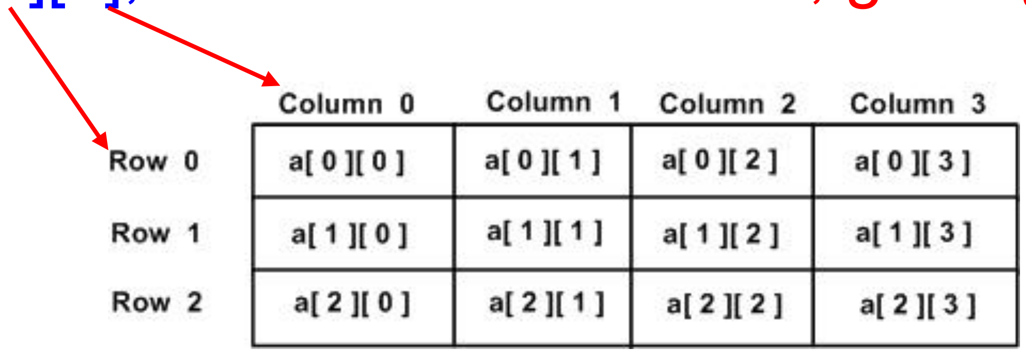
2D Arrays and Pointers

2D Arrays

- Declare a two-dimensional array of size [x][y]

  type arrayName [x][y];

int a[3][4]; /* not initialized, garbage value */



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

2D Arrays

- Initializing an array

```
int a[2][3] = {
  {1, 2, 3}, /* initializers for row indexed by 0 */
  {4, 5, 6}  /* initializers for row indexed by 1 */
};
```

- Nested braces are optional

```
int a[2][3] = {1,2,3,4,5,6};
```

Accessing 2D Arrays

```
#include <stdio.h>
#define n 2
#define m 3
int main (){
  int a[n][m] = {{1, 2, 3},{4, 5, 6} };
  int i, j;
  for ( i = 0; i < n; i++ ) {
    for ( j = 0; j < m; j++ ) {
      printf("a[%d][%d]=%d\n",i,j,a[i][j]);
    }
  }
  return 0;
}
```

Passing 2D Arrays as Parameters

```
#include <stdio.h>
#define N 2
#define M 3
void print(int [][M]);
void print(int (*)[M]);

int main (){
   int a[N][M] =
      {{1, 2, 3},
      {4, 5, 6} };
   print(a)
   return 0;
}
```

```
void print(int b[N][M])
{…}

void print(int b[][M])
{…}

void print(int (*b)[M])
{…}
```

Passing 2D Arrays as Parameters

```
#include <stdio.h>
#define N 2
#define M 3
void print(int **);

int main (){
   int a[N][M] =
      {{1, 2, 3},
      {4, 5, 6} };
   print(a)
   return 0;
}
```

```
void print(int b[N][M])

void print(int b[][M])

void print(int (*b)[M])
```

```
void print(int **b)
{…}
```

Incorrect

2D Arrays in Memory

- 2D array is stored sequentially in memory in row major order.

```
int a[2][3] = {
 {1, 2, 3},
 {4, 5, 6}
};
```

- Is stored in memory as

a	**1**	**2**	**3**	**4**	**5**	**6**
	`a[0][0]`	`a[0][1]`	`a[0][2]`	`a[1][0]`	`a[1][1]`	`a[1][2]`

Accessing 2D Arrays

```
#include <stdio.h>
int main (){
  int a[2][3] = {{1, 2, 3},{4, 5, 6, 7}}
  int *p = a;
  for ( i = 0; i < 2 * 3; i++ ) {
     printf("%d\n", *p++);  /* p[i] */
  }
  return 0;
}
```

Accessing 2D arrays using Pointers

- Given a 2D array:

```
int a[m][n];;
```

- To find a[0][2], we do the following:

```
*(a[0] + 2)    /* same as a[0][2]  */
```

- In general:

```
a[i][j] = *(a[i] + j)
```

Accessing 2D arrays using Pointers

- Given a 2D array:

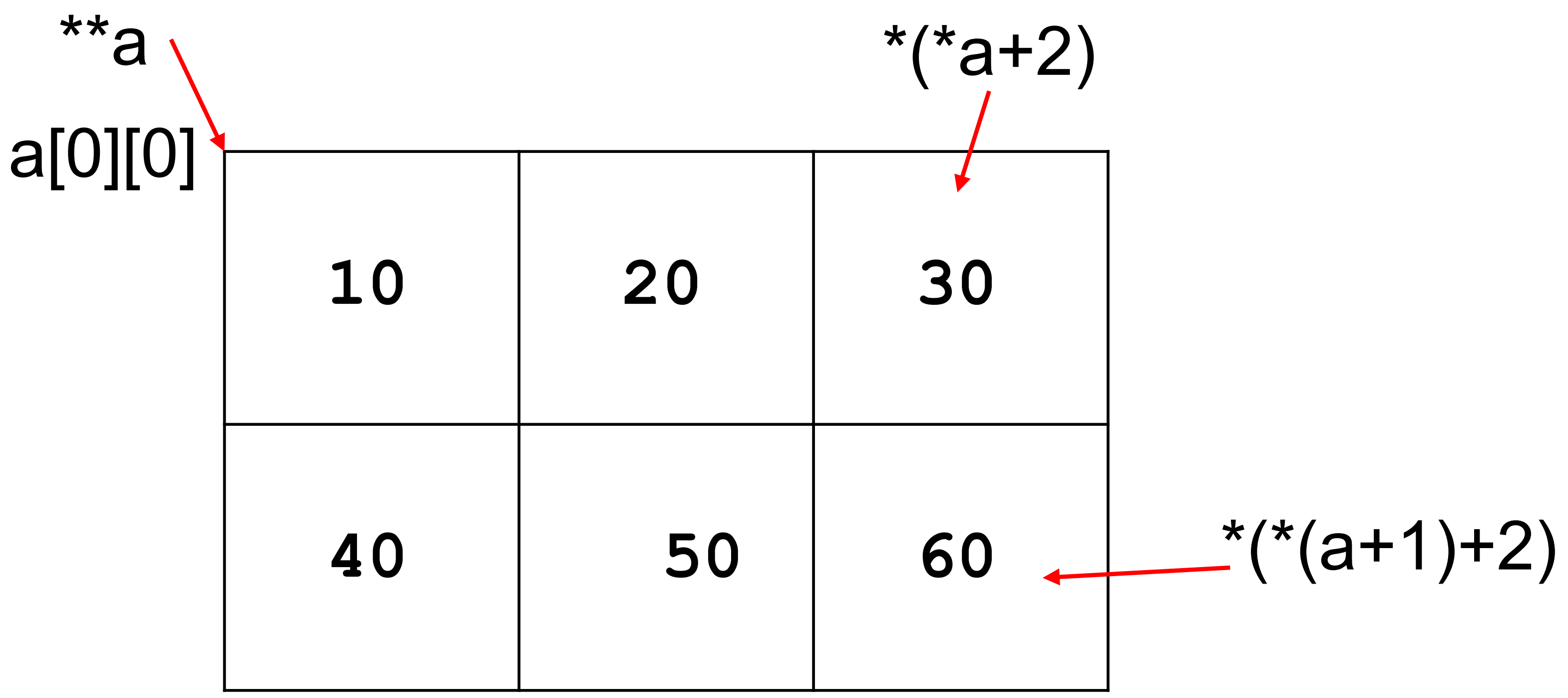
```
int a[m][n];;
```

- Because `a[0] = *a`

```
a[i][j] = *(a[i] + j) = *(*(a+i) + j)
```

Accessing 2D arrays using Pointers

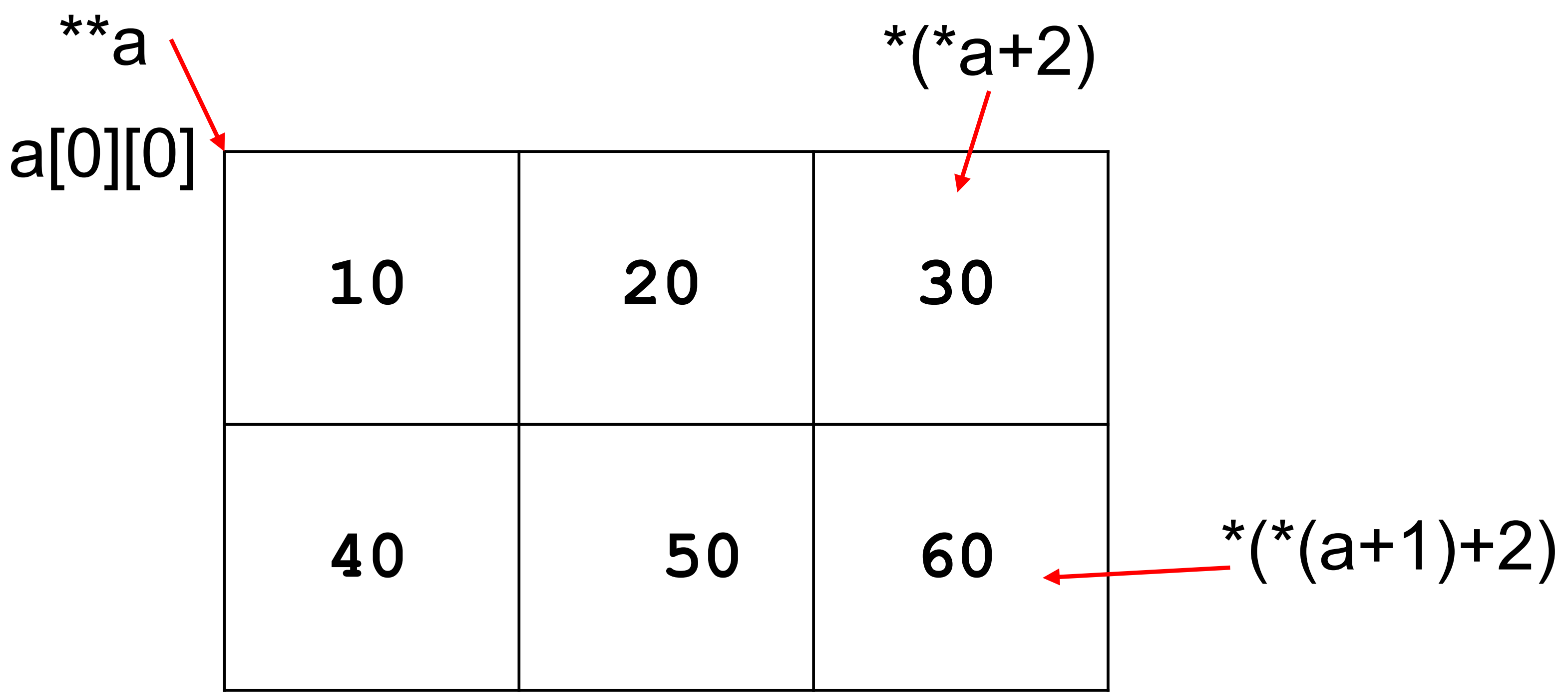
- `int a[2][3]={{10, 20, 30},{40, 50, 60} };`



```
a[i][j] = *(*(a+i) + j)
```

Accessing 2D arrays using Pointers

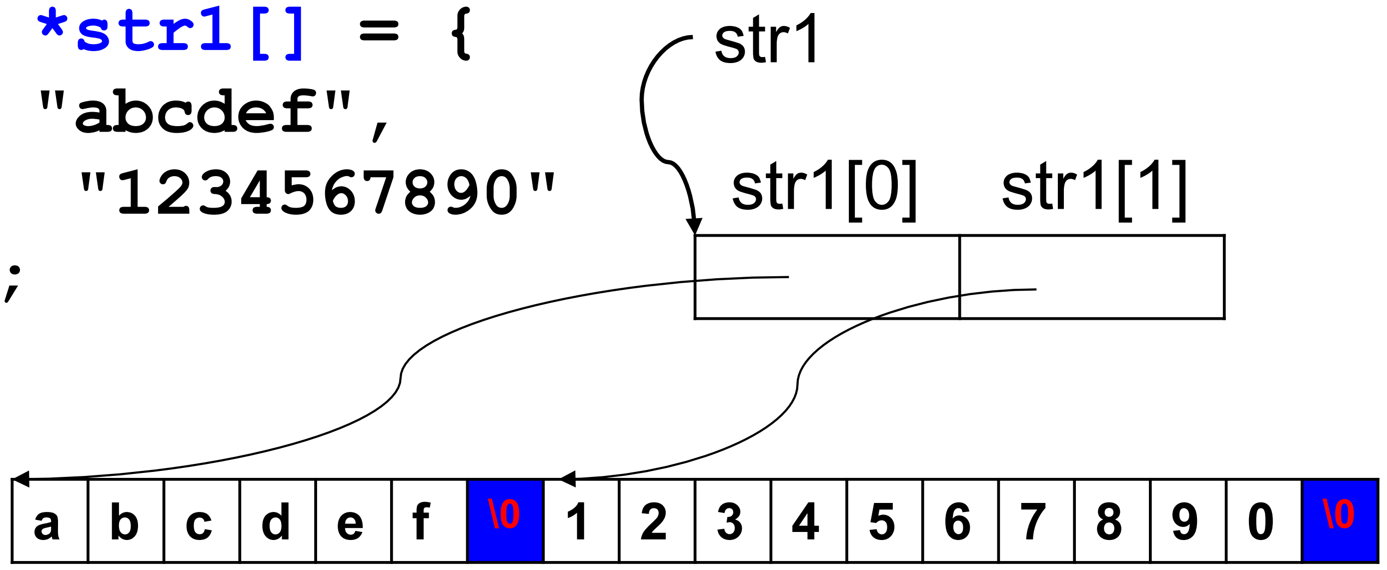
- `int a[2][3]={{10, 20, 30},{40, 50, 60} };`



```
a[i][j] = *(*(a+i) + j)
```

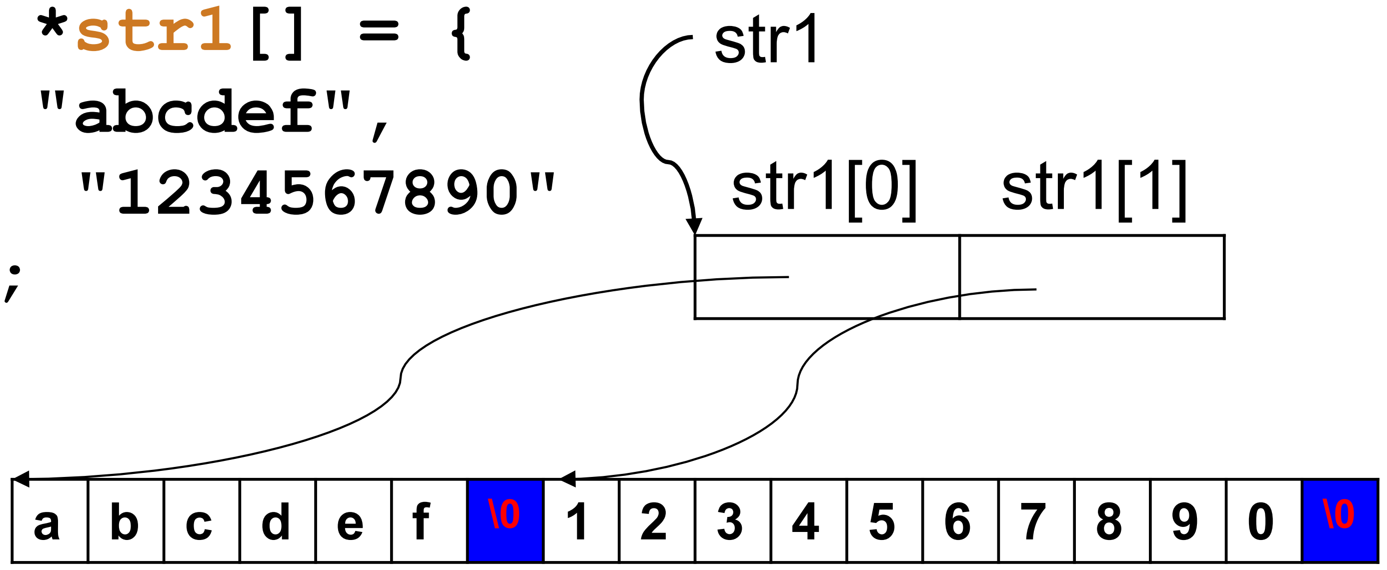
String Array Pointers

```
char *str1[] = {
      "abcdef",
       "1234567890"
   };
```

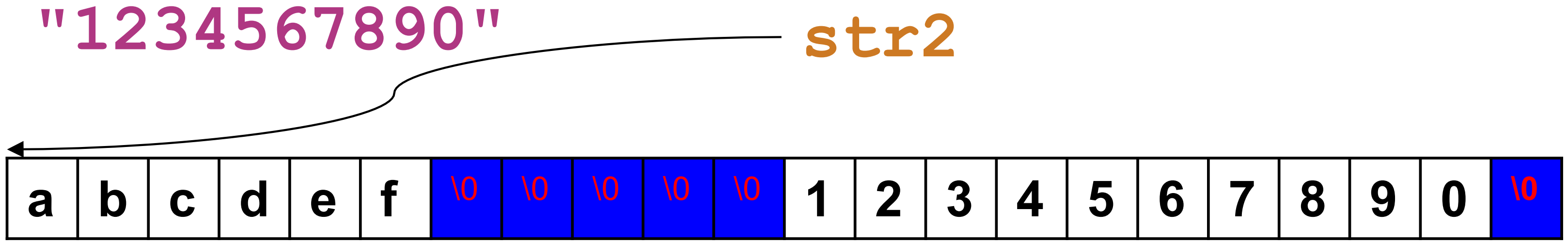


String Array Pointers

```
char *str1[] = {
      "abcdef",
       "1234567890"
   };
```



```
char str2[][11] = {
      "abcdef",
       "1234567890"
   };
```



String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
printf("%d\n", sizeof(str1[0]));
printf("%d\n", sizeof(str2[0]));
```

String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
printf("%d\n", sizeof(str1[0]));  8  pointer
printf("%d\n", sizeof(str2[0]));   11
```

String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
printf("%d\n", strlen(str1[0]));
printf("%d\n", strlen(str2[0]));
```

String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
printf("%d\n", strlen(str1[0]));     6
printf("%d\n", strlen(str2[0]));     6
```

String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
Print memory address:
&(str1[0][0]);      8666
(str1[0]));         8666
(str1[1]));         8673
(str2[0]));         7216
(str2[1]));         7227
```

String Array Pointers

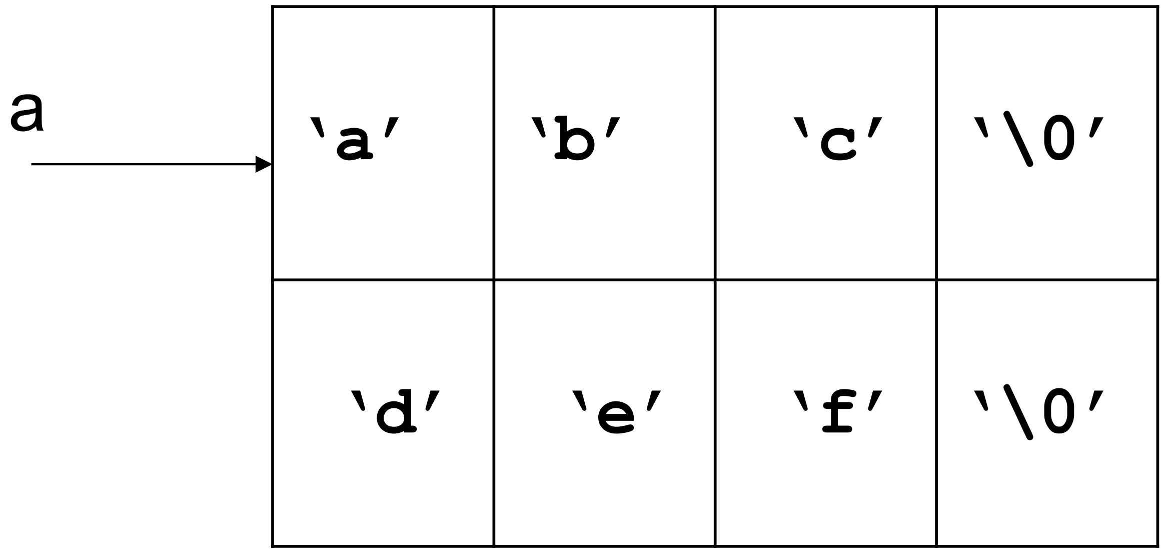
```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
char *p = (char *) (str1[0]);
for(int i = 0; i < 18; i++){
     printf("%d,",*p++);
}
97,98,99,100,101,102,0,49,50,51,52,53,54,55,
56,57,48,0,
```

String Array Pointers

```
char *str1[] = {
     "abcdef",
      "1234567890"
   };
char str2[][11] = {
     "abcdef",
      "1234567890"
   };
char *p = (char *) (str2[0]);
for(int i = 0; i < 22; i++){
     printf("%d,",*p++);
}
97,66,99,100,101,102,0,0,0,0,0,49,50,51,52,5
3,54,55,56,57,48,0
```

String Arrays

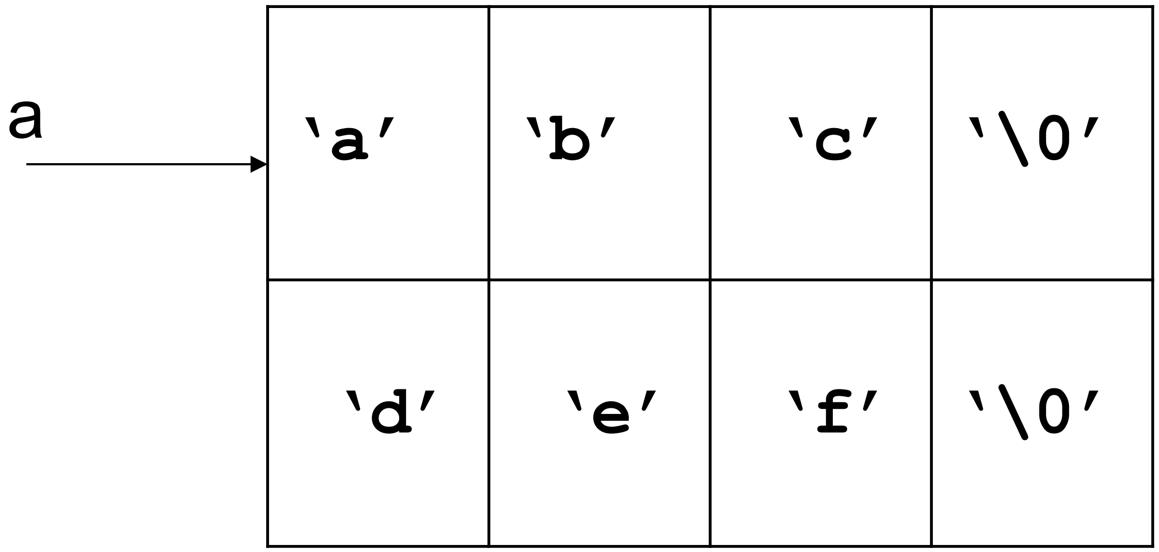
- `char a[2][4]={"abc","def"};`



```
printf("%s\n", a[0]);
a[0][3] = '1'
printf("%s\n", a[0]);
```

String Arrays

- `char a[2][4]={"abc","def"};`



```
printf("%s\n", a[0]);
```
abc

```
a[0][3] = '1'
printf("%s\n", a[0]);
```
abc1def