char p[10] = "goat";

1. declaration: name + type of a variable

2. memory allocation: Set ("type") aside 10 bytes for 10 chars.

3. Initialization: put some data into those bytes.
char p[10];
just declaration + mem alloc

char *p;
declaration + mem alloc

no initialization yet.

8 bytes to hold the pointer
char *p = "horse";
dcl.
allocates 8 bytes
allocates 6 bytes
initializes p to point to "horse"
Char * p = "horse"

Char * p = "horse"

Strcpy(p, "goat");

Crashes
You're trying to write
goat into read-only
memory
char *p = "horse";
printf("%c", p[2]);

compiler arranges:
*(p + 2 * sizeof(char)) → r

on screen
Quiz #4

$k = 0xfffffffd6$

little-endian
$0 	imes \text{fffff} \text{ff} \text{dd}k = 32$-bit 2's comp \(-42\)

<table>
<thead>
<tr>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{32} & + \text{8} + \text{2} = 42
\end{align*}
\]
More pointer stuff

```c
int a[4] = {3, 9, 2, 5};
int *p = a;

printf("%0x", a) \rightarrow 3
printf("%0x", p) \rightarrow 3
printf("%0x", a[0]) \rightarrow 3
printf("%0x", &p) \rightarrow 8880
printf("%0x", &a) \rightarrow 3
```
\[ \text{Int } \times [47]\]

\[ \text{size of } (a) \rightarrow 4 \times 16 \]

\[ \text{Int } \ast p = a \]

\[ \text{size of } (p) \rightarrow 8? \]
Mixed-type arrays?

Python

\[ a = ['goat', 6] \]

Not in C

\[ \text{void} *a[100] ; \]
int b = 0x08005432;
char c = (char)b

\[ \text{addr: } 0x32 \]