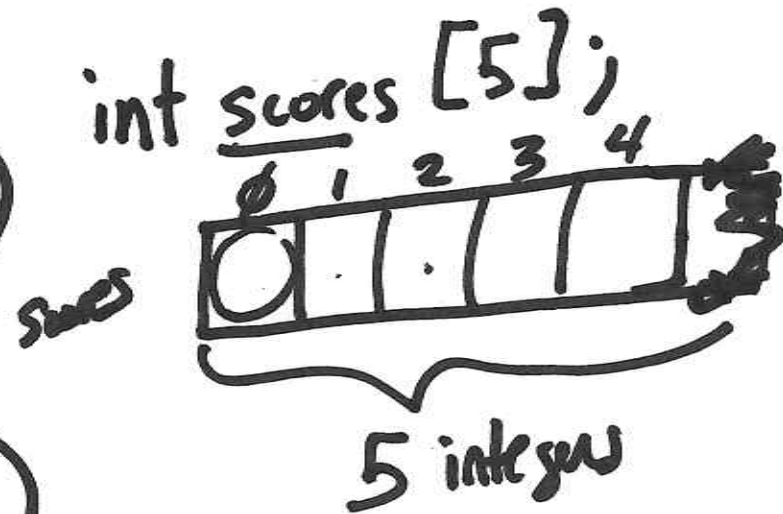


```
int hw1; ~□  
int hw2; ~□  
⋮  
int hw5; ~□
```

```
cin >> hw1;
```



index ← access an element of an array

```
cin >> scores [0];
```

100
90
90
100
95

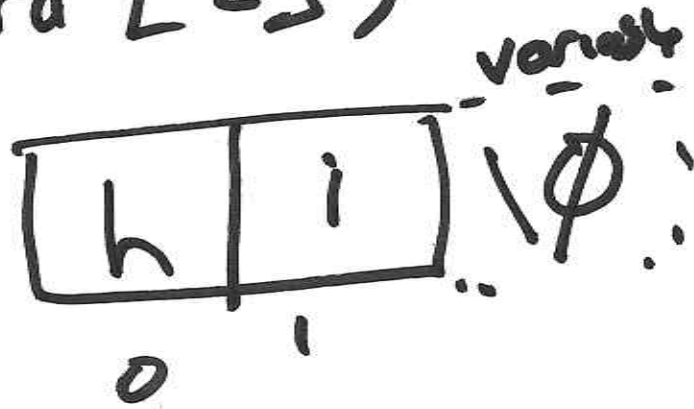
```
int scores [5];  
cout << "please enter all 5 scores";  
for (int i = 0; i < 5; ++i)  
{  
    cin >> scores[i];  
}
```

i [0] 1 2 3 4 5

↑
subscript

100	90	90	100	95
int	int	int	int	int
0	1	2	3	4

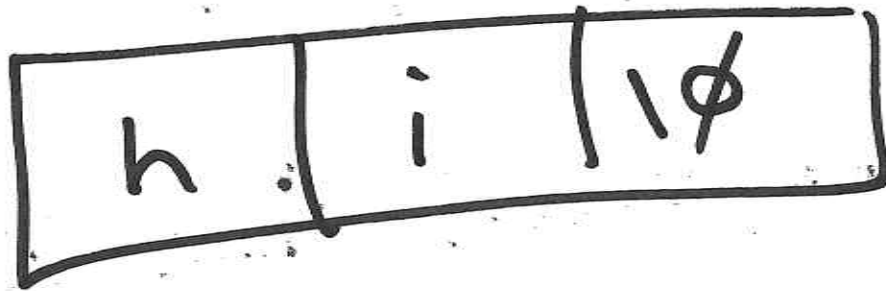
char word [2];
int count



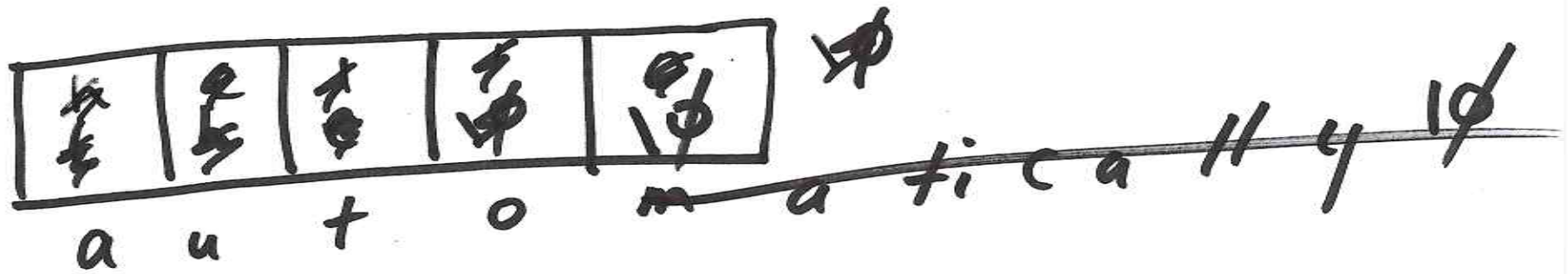
cin >> word; //read in a word

char word [3]

↑
minimum



```
char word [5];  
cin.width(5); cout << "please enter a word";  
cin >> word;
```



SEG FAULT

```
const int SIZE = 21;
```

```
int main()  
{
```

```
    char fname [ SIZE ];
```

```
    cin.width ( SIZE );
```

```
    cin >> fname;
```

```
const int SIZE = 131;
```

```
int main()
```

```
{
```

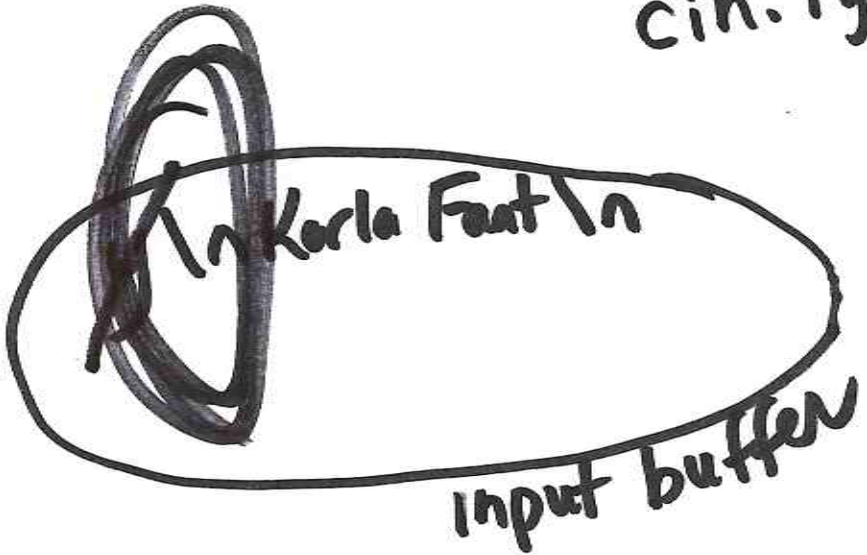
```
    char line [SIZE];
```

```
    cout << "Enter in a line of text....";
```

```
    cin.get(line, SIZE, '\n');
```

```
char response;  
char name [21]; Karla Fant \n  
cout << "do you want to read in a name?"  
cin >> response; cin.ignore (100, '\n');
```

```
if ( response == 'y' )  
{  
    cout << "Enter your name";  
    cin.get (name, 21, '\n');  
    cin.ignore (100, '\n');
```



Every Time You Read
(Anything) - Remove the
delimiter (i.e., newline)

Line

char line[131];

cin.get(line, 131, '\n');

Sentence

cin.get(line, 131, '\n');
1 character

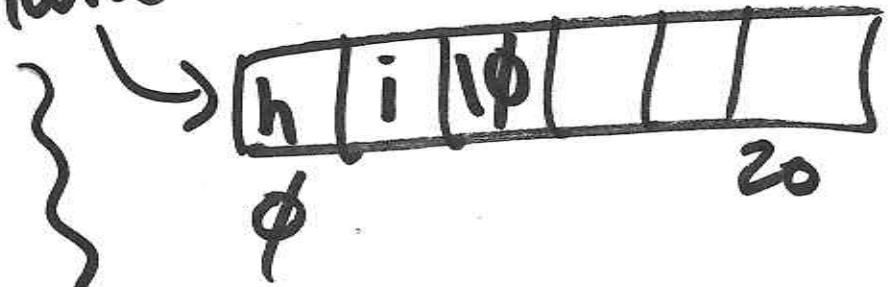
parag

char parag[50];

cin.get (parag, 50, '#')
cin.ignore (100, '\n'); // Remove #
and enter



```
Char name1[21];  
Char name2[21];
```



```
if (name1 == name2)  
    comparing locations
```

WRONG
strcmp
strcpy
strlen

```
cout << name1;
```