

1. How would you make sure only 40 characters get stored in an array when we do input:
char array[41];

cin >> array;
Read 1 word
of any size

① cin.width(41)
↑ size of array
Limits what is read to 40

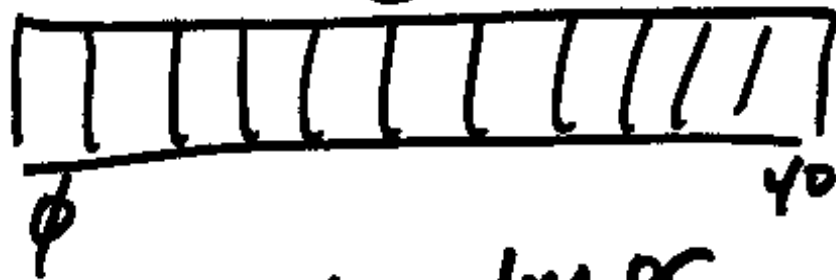
or
② cin.get(array, 41, '\n');
↑ optional
Read in multiple words

2.

2. what is wrong with:

`c.in.get(array, 40, '\0');`

Always
Read
40
characters



input buffer

hello world\n
this is really\n
fun\n

Read until 40 characters or
~~the~~ delimiter is encountered
but not read. The delimiter
is 1 character and it MUST
be something the user at
the keyboard can type.

call

```
int main()
{
    int answer;
    answer = sum(3, 10);
}
```

```
void sum(int, int, int &);
```

```
void sum(int x, int y, int &result)
```

```
{  
    result = x + y;
```

$x + y$

```
}
```

```
int main()
```

```
{  
    int answer;
```

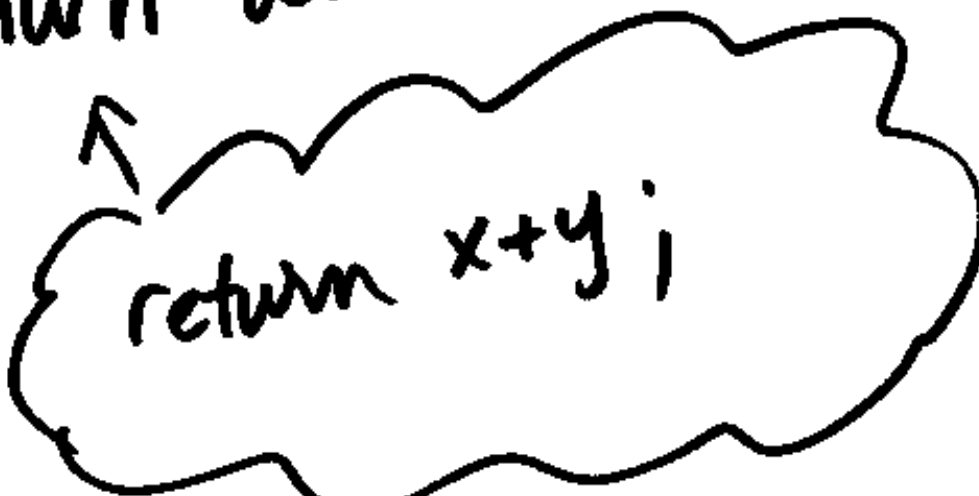
13

```
answer = sum  
sum(3, 10, answer);
```

- // Write a function that will
- // Sum together two values (integers)
- // & supply the answer back

Answer → int sum (int x, int y)

{ int total = x + y;
return total

}


return x+y;

main

```
int sumup = 0; int num;  
// Create a Running Total
```

```
do  
{ cout << " please enter a number "  
    " enter negative to stop "
```

```
cin >> num;
```

```
if (num >= 0)
```

```
sumup = sum(sumup, num);
```

```
} // end when negative
```

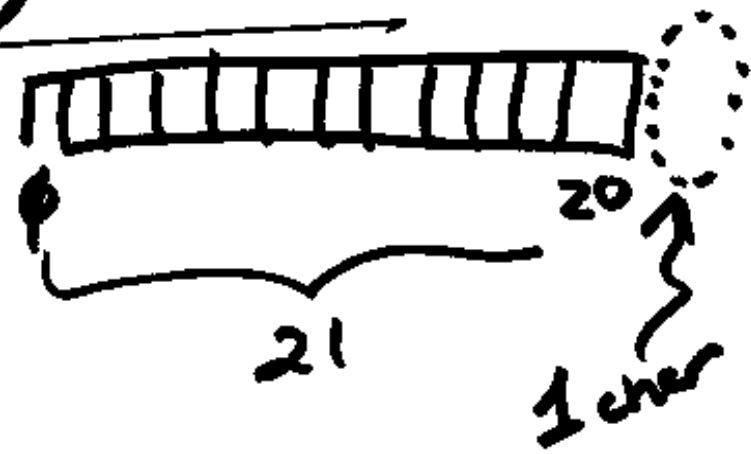
```
while (num >= 0);
```

Valid / invalid?
- "bad"

Given:

```
char name [21];  
char name2 [41];
```

```
cin >> name [21];  
cin >> name [0];  
cin >> name;  
name [0] = '\0';  
cout << name;  
cout << name [0];  
if (name == name2)
```



- (a)
- ✓ (b)
- ✓ (c)
- ✓ (d)
- ✓ (e)
- (f)
- (g)

Write a C++ statement
to store "Spring Break"
into a variable
strcpy (phrase, "Spring Break");

// Arrays of characters - For Reading

void
{

read_confirm(char prompt[], char data[],
int size)

do
{
cout << prompt << " : ";
cin.width(size);
cin >> data; cin.ignore();
cout << "Is this correct? Y/N";
cin >> response; cin.ignore();
} while (response != 'n');

int size

}



char name[21];

read_confirm("Enter ^{First} ~~the~~ Name", name, 21);

main