

memory

address:

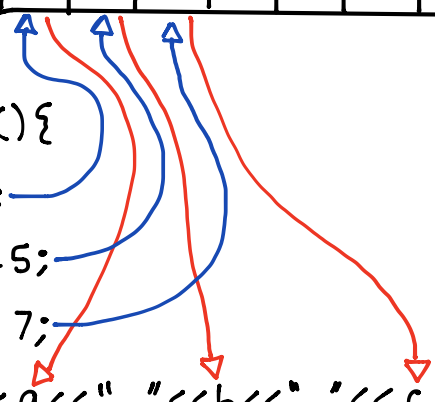
value:

[illegible]

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:	?	5	7	?											?

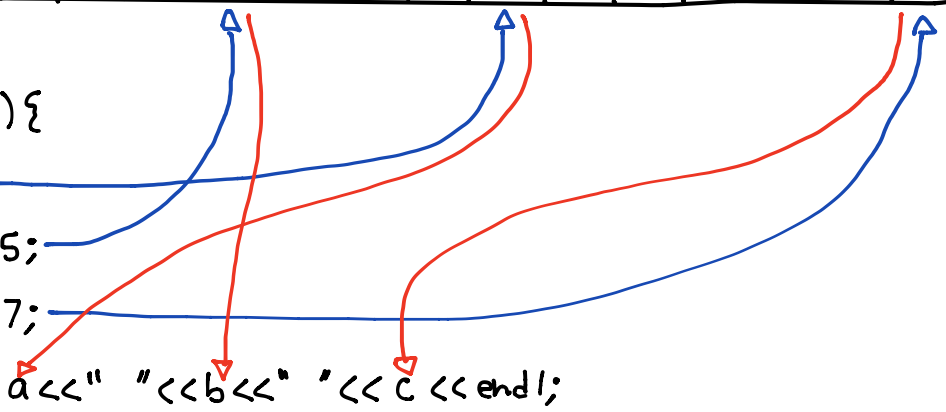
```
int main(){  
    int a;  
    int b=5;  
    int c=7;  
    cout<<a<<" "<<b<<" "<<c<<endl;  
    return 0;  
}
```



memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:	?	?	5	?	?	?	?	?	?	?	?	?	7	?	

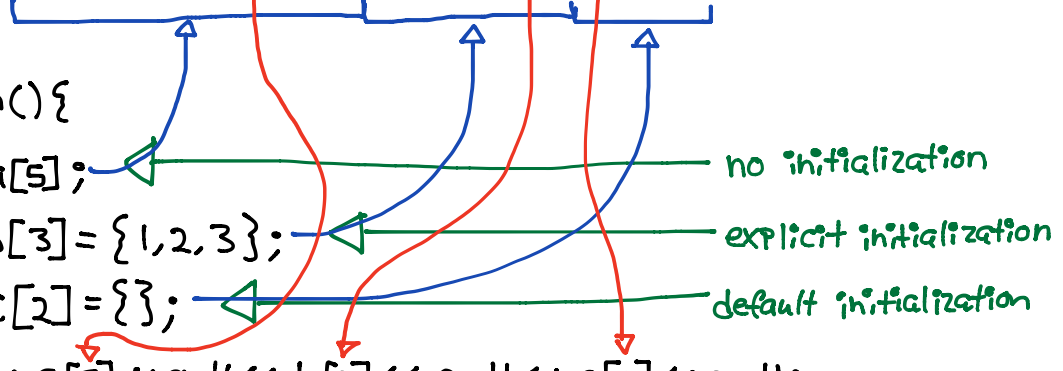
```
int main(){  
    int a;  
    int b=5;  
    int c=7;  
    cout<<a<<" "<<b<<" "<<c<<endl;  
    return 0;  
}
```



memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:	?			?	1	2	3	0	0	?					?

```
int main(){
    int a[5];
    int b[3] = {1, 2, 3};
    int c[2] = {};
    cout << a[3] << endl << b[2] << endl << c[0] << endl;
}
```



memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:	?	—	?	?	—	?	?	1	3	5	0	0	?	?	—

```
int main(){
```

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

```
    int b[2];
```

```
    int c[3];
```

```
}
```

if only some of the values are explicitly initialized, the rest of them will be initialized to whatever the default value is for that type (zero, for integer types)