CS101: Introduction to Programming

Venkatesh Vinayakarao

Department of Computer Science and Engineering

IIIT Sri City, Chittoor.

venkatesh.v@iiits.in

Online Judges

- Please try https://uva.onlinejudge.org/index.php
- Solve problem **136 - Ugly Numbers**
- No need to submit anything. Not graded.

Bonus Task – 4% Marks

- Applies only if the total (after including the bonus) < 90%.
- Only top-6 submissions by quality will get the bonus marks.
- Instructor judgment is final.
- Deadlines are strict.

Bonus Task 1

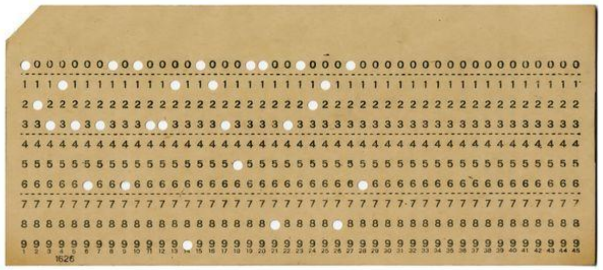
- Make a 5 to 10 slide presentation on Compilers
- Your 15 minute presentation should cover all (but not limited to) the following topics:
    - Variety of C compilers
    - What does a C compiler do?
    - Give the history of C compilers.
    - An example of code which gives different output in different compilers
    - Which compiler should I use and why?
- Deadline: 18th Sep 2018.

Agenda

- Pointers
  - Introduction

History of Computers

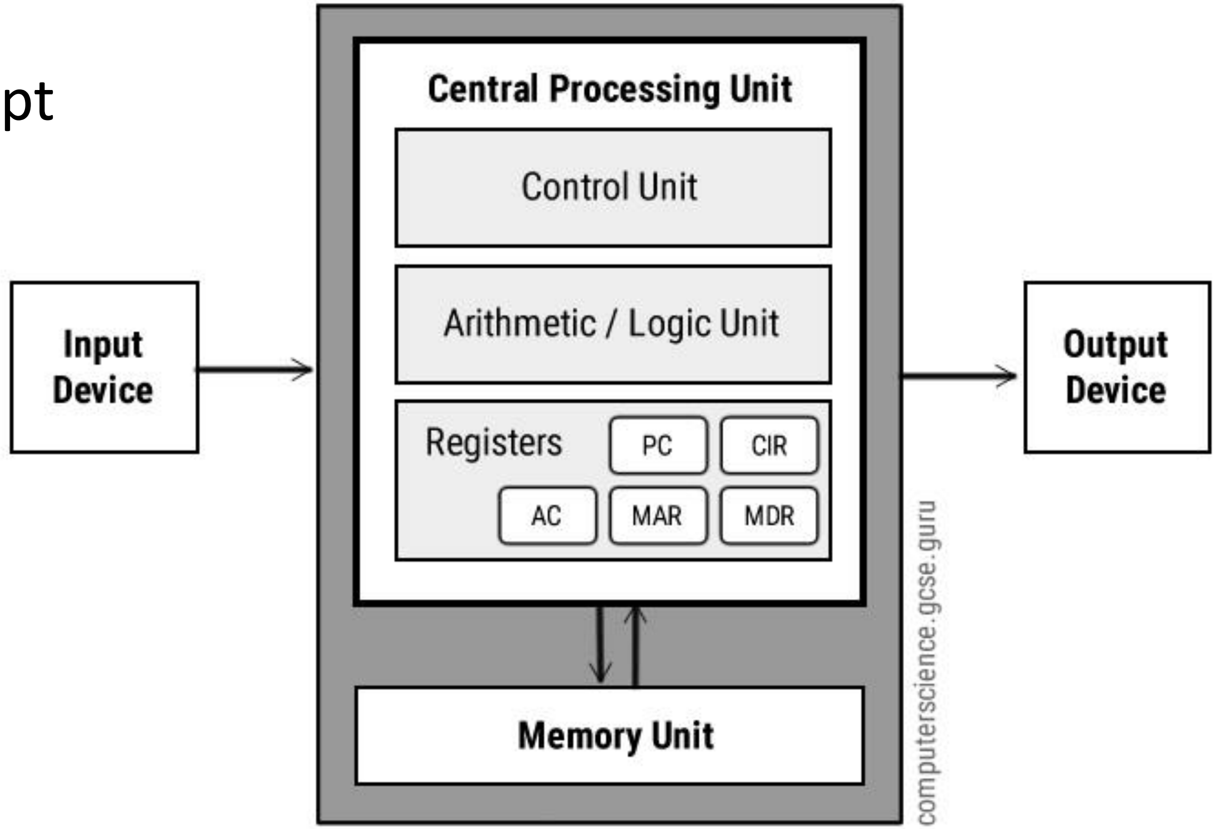
- Earlier computers could not store data or programs.
- Punch Cards were used to store data and programs as early as 1725.
  - In 1890, Herman Hollerith developed machines to read and write punch cards.
  - He later formed IBM.
  - Thus, punch cards are also known as Hollerith Cards or IBM Cards.



ComputerHope.com

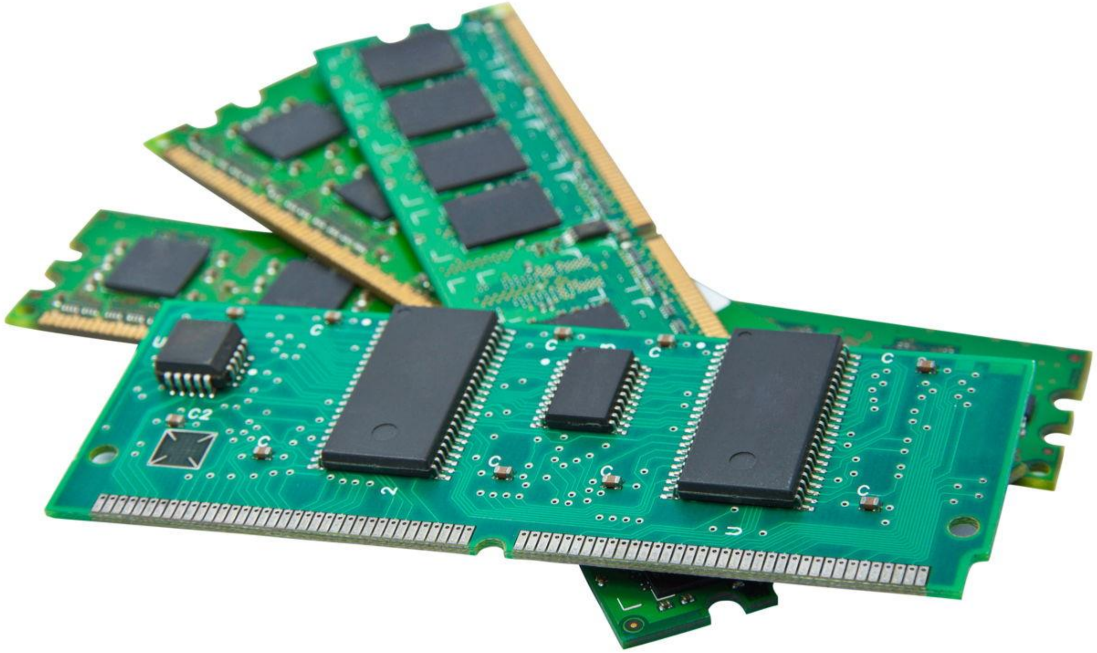
How does it work?

- Von Neumann Architecture
  - 1940 – Stored Program Concept



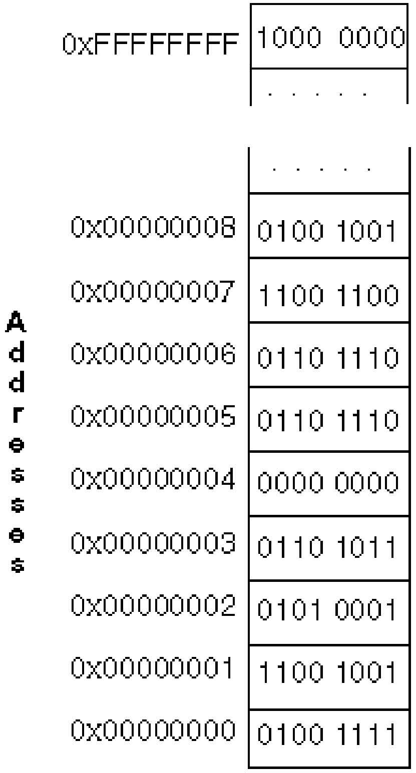
computerscience.gcse.guru

Computer Memory



Memory and Addresses

- Computer with 1 GB RAM has an array of 1024 * 1024 * 1024 Bytes.

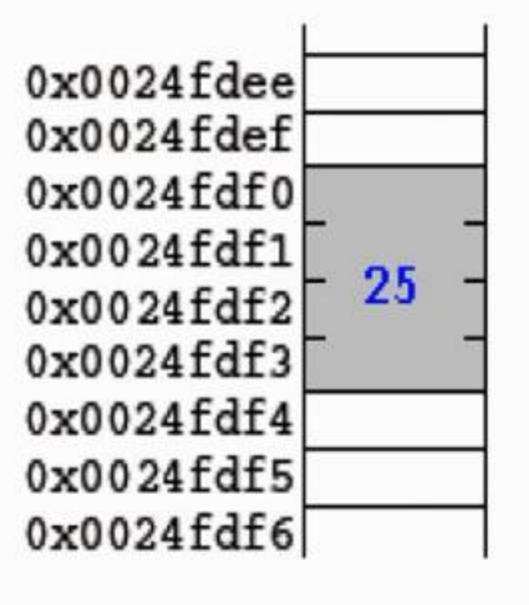


0xFFFFFFFF	1000 0000
	. . . . .
	. . . . .
0x00000008	0100 1001
0x00000007	1100 1100
0x00000006	0110 1110
0x00000005	0110 1110
0x00000004	0000 0000
0x00000003	0110 1011
0x00000002	0101 0001
0x00000001	1100 1001
0x00000000	0100 1111

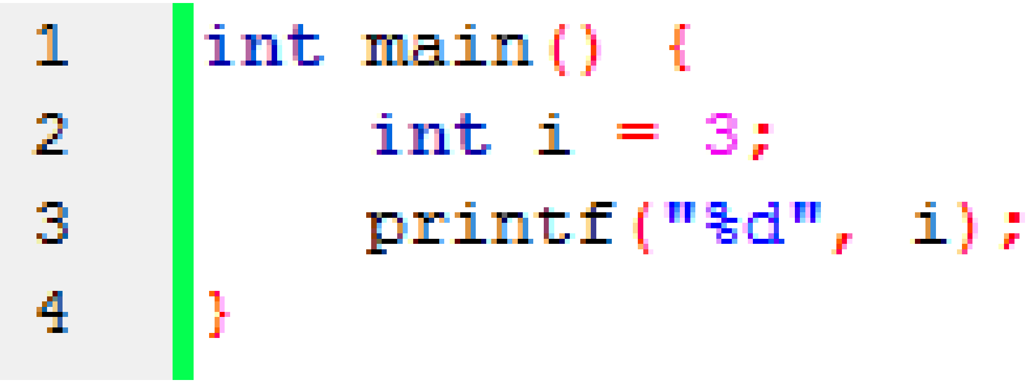
Main Memory

Variable Declaration

- What happens when int x = 25; is run?
  - Depending on the data type (int in this case), some amount of memory is reserved.
  - The value 25 is written to this memory location.

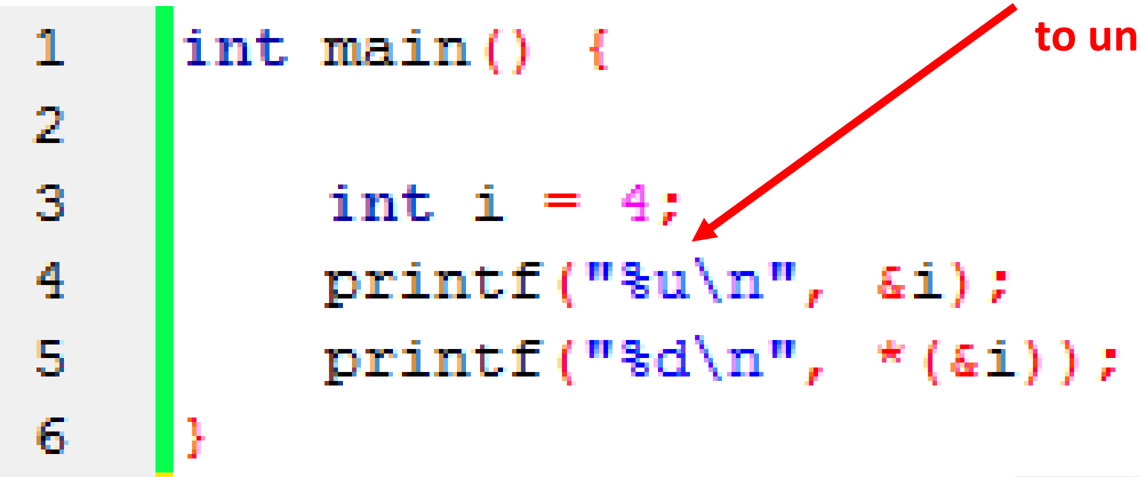


3 is printed…



```c
int main() {
    int i = 3;
    printf("%d", i);
}
```

Where is i stored?



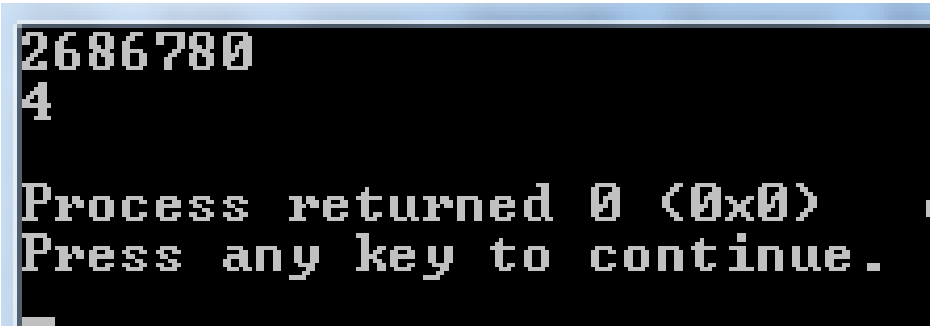
```c
int main() {

    int i = 4;
    printf("%u\n", &i);
    printf("%d\n", *(&i));
}
```

Memory address cannot be negative. %u refers to unsigned int.

&i refers to address of i.

*(&i) refers to the value at the address of i.

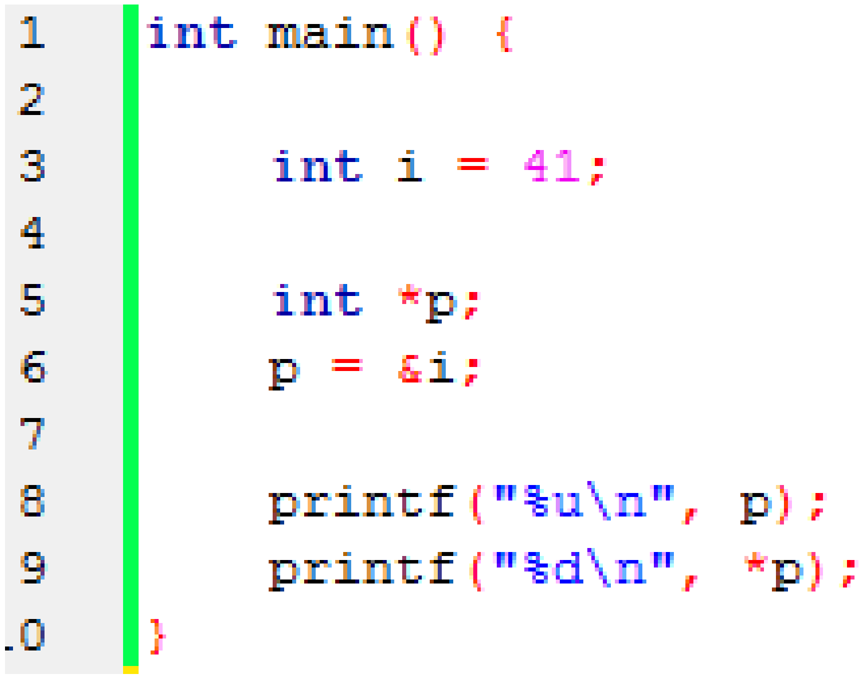


```
2686780
4

Process returned 0 (0x0)
Press any key to continue.
```

A Pointer Variable

- int *p is a declaration of a pointer variable p.
- We say that p points to some location.

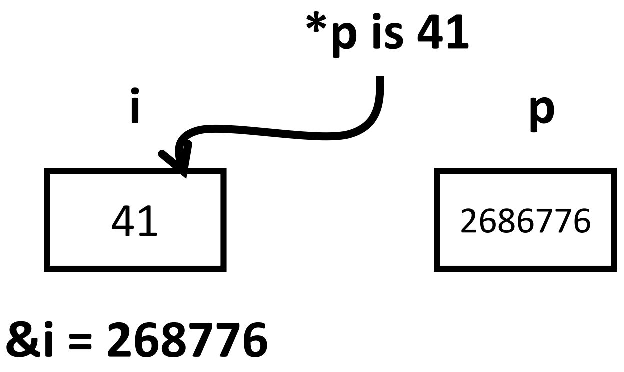


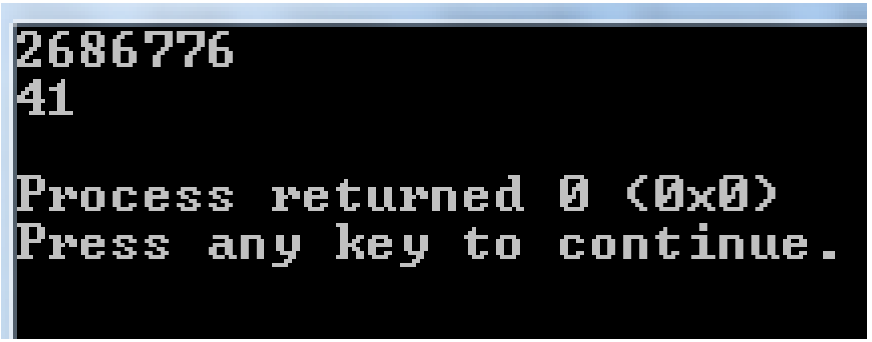
```
int main() {

    int i = 41;

    int *p;
    p = &i;

    printf("%u\n", p);
    printf("%d\n", *p);
}
```



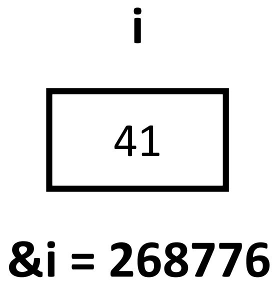


```
2686776
41

Process returned 0 (0x0)
Press any key to continue.
```

Pointers

- What is the value of i?
- What is the address of i?



i

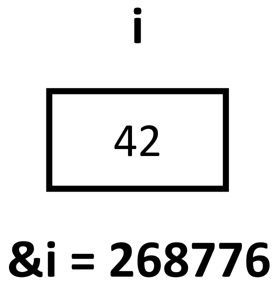
41

&i = 268776

A memory snapshot while running your program

Quiz

- What is the value of i?
- What is the address of i?



i

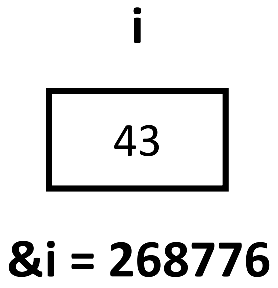
42

&i = 268776

A memory snapshot
while running your
program

Quiz

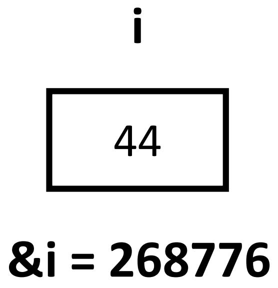
- What is the memory snapshot of int i = 43; ?



A memory snapshot
while running your
program

Quiz

- What is the memory snapshot of int i = 44; ?



i

44

&i = 268776

A memory snapshot
while running your
program

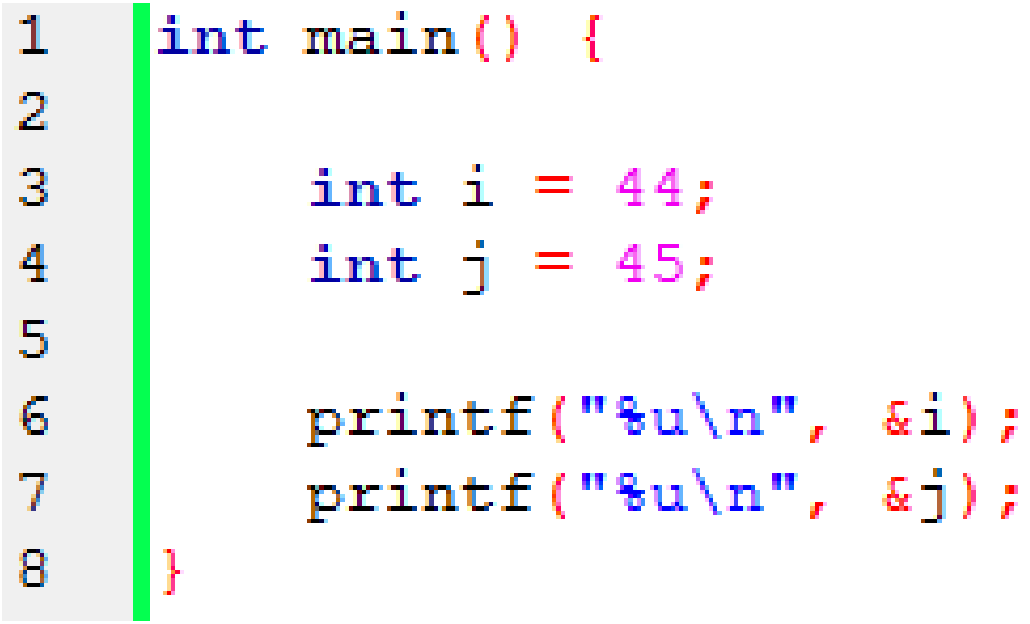
Quiz

- What is the memory snapshot of the following code?

```
int i = 44;
int j = 45;
```

Memory and Variables

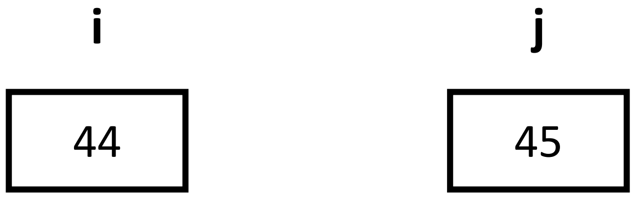
What is the memory snapshot of the following code?



```c
int main() {

    int i = 44;
    int j = 45;

    printf("%u\n", &i);
    printf("%u\n", &j);
}
```

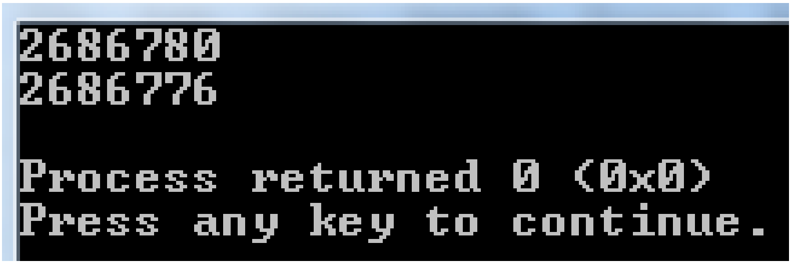


&i = 2686780

&j = 2686776

A memory snapshot while running your program

A memory snapshot while running your program



```
2686780
2686776

Process returned 0 (0x0)
Press any key to continue.
```

Quiz

- What is the memory snapshot of the following code?

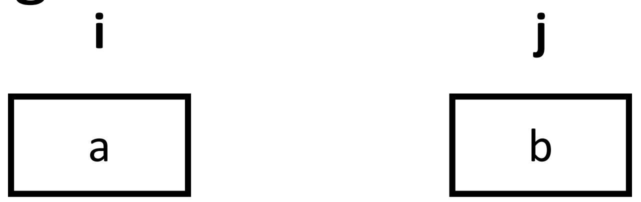
```
char i = ‘a’;
char j = ‘b’;
```

Quiz

- What is the memory snapshot of the following code?

char i = ‘a’;

char j = ‘b’;



i

a

j

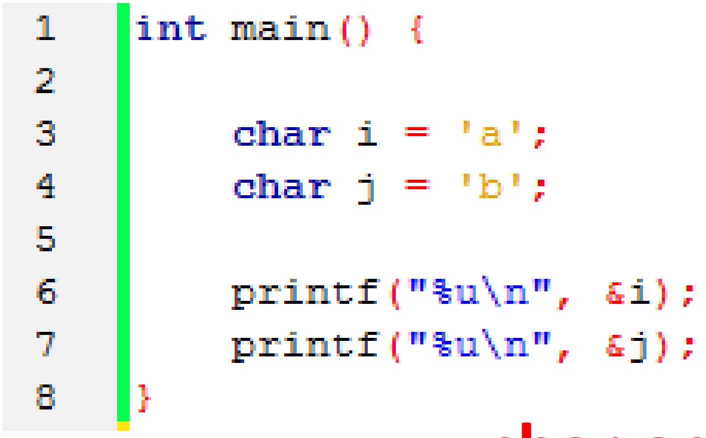
b

&i = 2686783

&j = 2686782

A memory snapshot while running your program

A memory snapshot while running your program

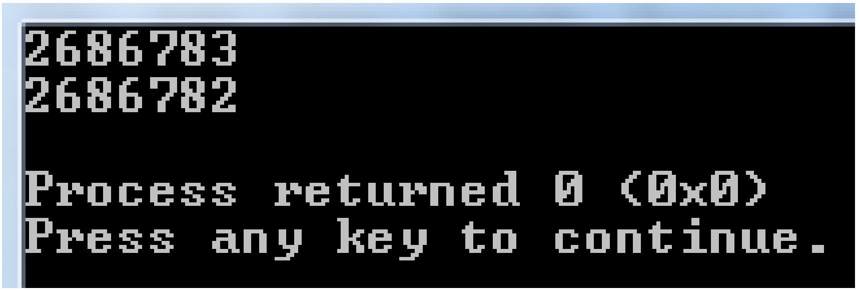


```
int main() {

    char i = 'a';
    char j = 'b';

    printf("%u\n", &i);
    printf("%u\n", &j);
}
```

char occupies only 1 byte!



```
2686783
2686782

Process returned 0 (0x0)
Press any key to continue.
```

Quiz

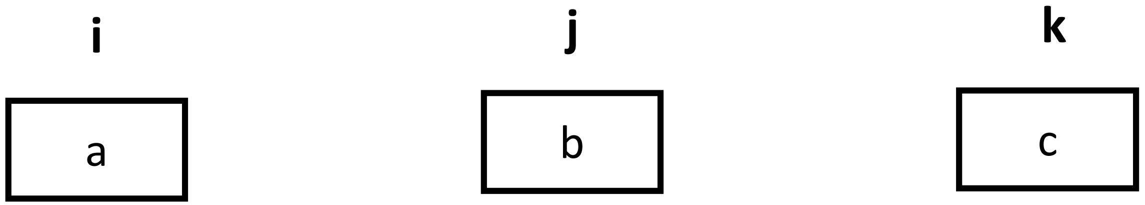
- What is the memory snapshot of the following code?

```
char i = ‘a’;
char j = ‘b’;
char k = ‘c’;
```

Quiz

- What is the memory snapshot of the following code?

```
char i = ‘a’;
char j = ‘b’;
char k = ‘c’;
```



i

a

&j = 2686783

A memory snapshot while running your program

j

b

&j = 2686782

A memory snapshot while running your program

k

c

&j = 2686781

A memory snapshot while running your program

Quiz

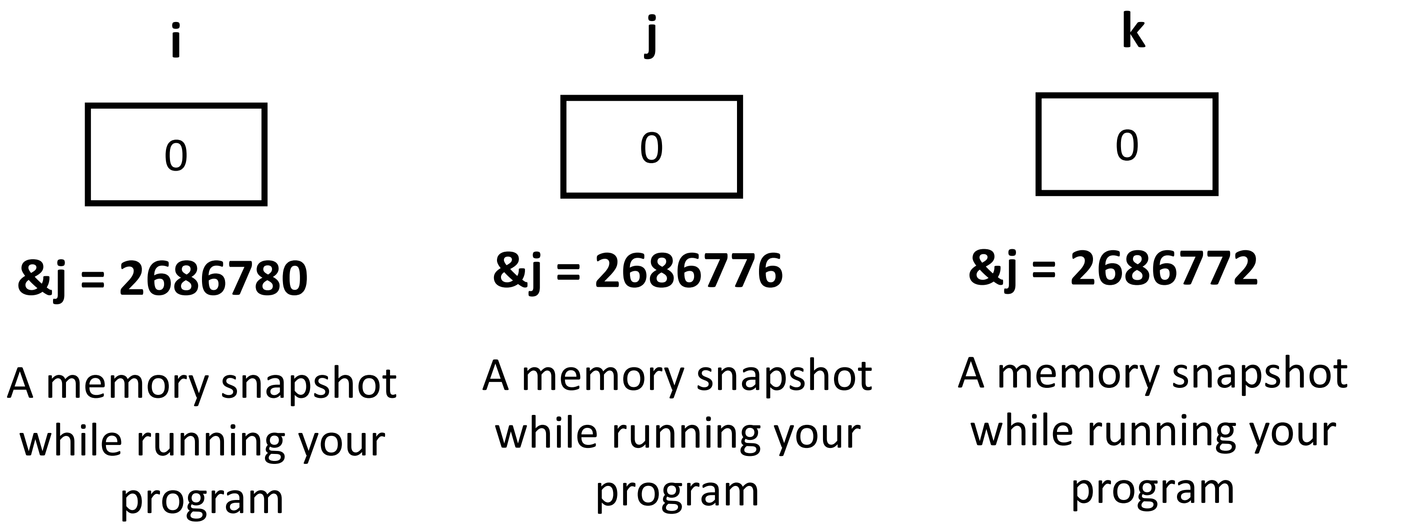
- What is the memory snapshot of the following code?

```
int i = 0;
int j = 0;
int k = 0;
```

Quiz

- What is the memory snapshot of the following code?

```
int i = 0;
int j = 0;
int k = 0;
```



i

0

&j = 2686780

A memory snapshot while running your program

j

0

&j = 2686776

A memory snapshot while running your program

k

0

&j = 2686772

A memory snapshot while running your program