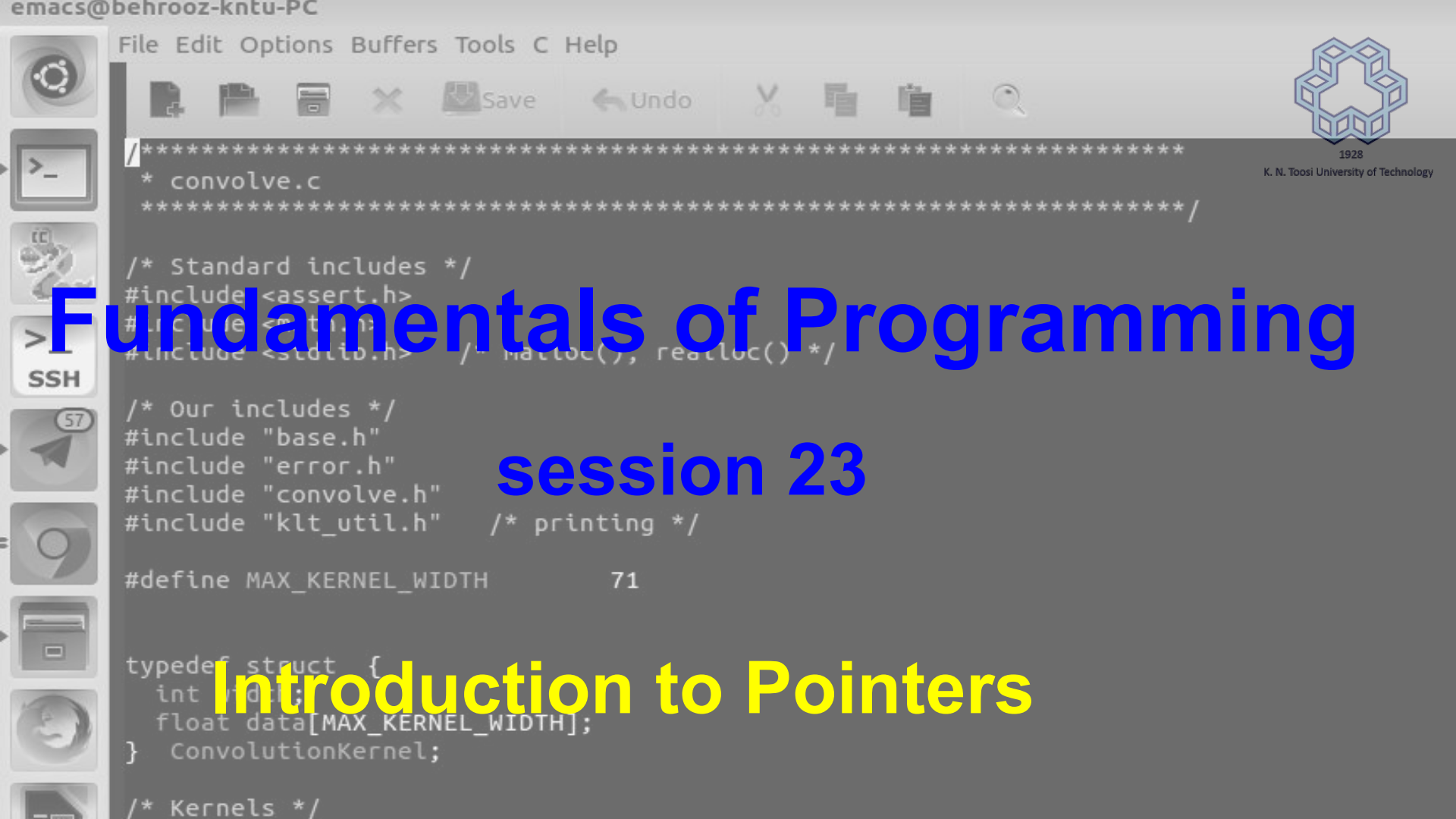
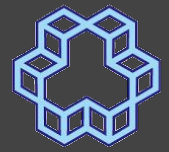


Fundamentals of Programming

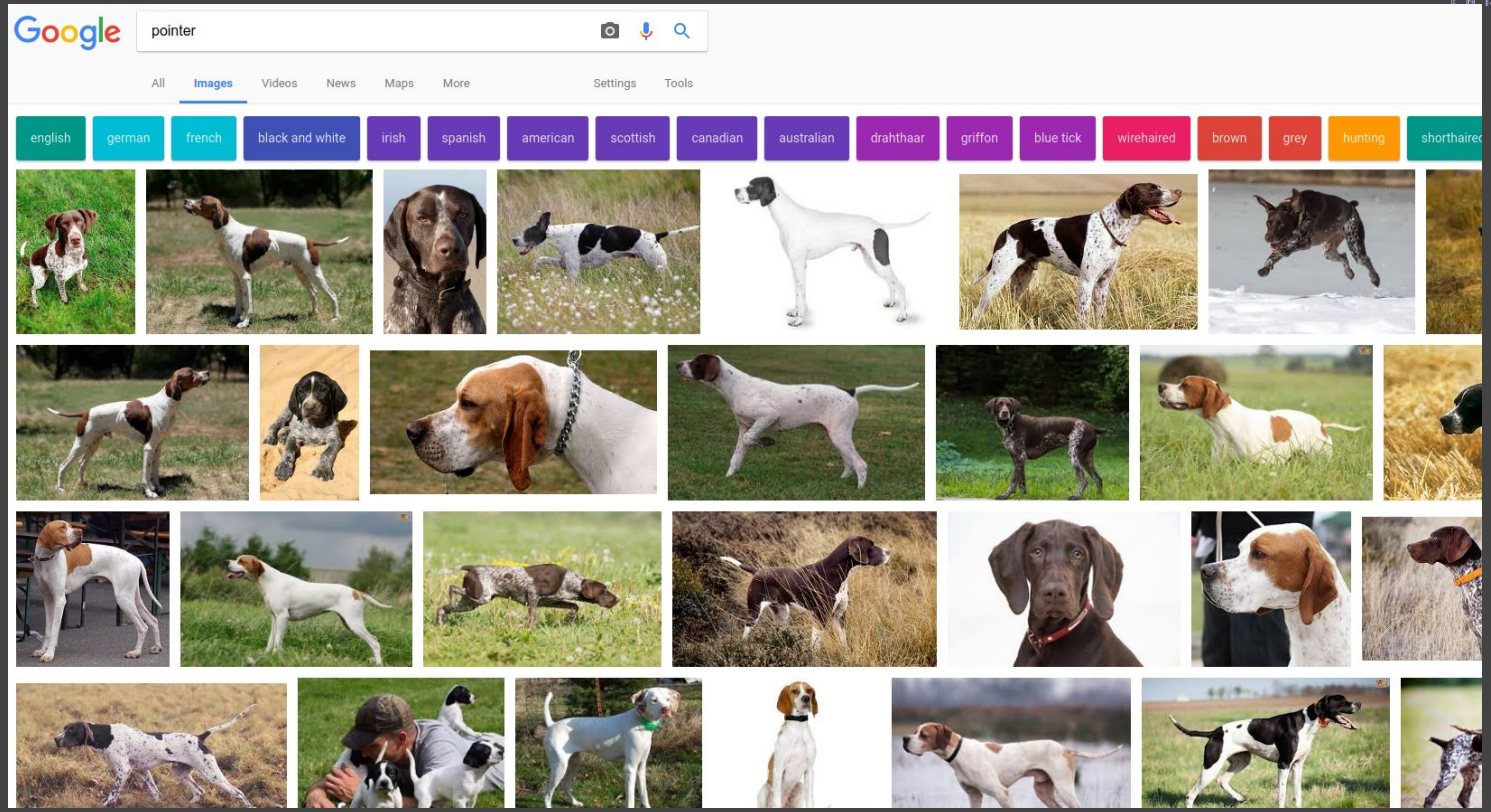
session 23

Introduction to Pointers

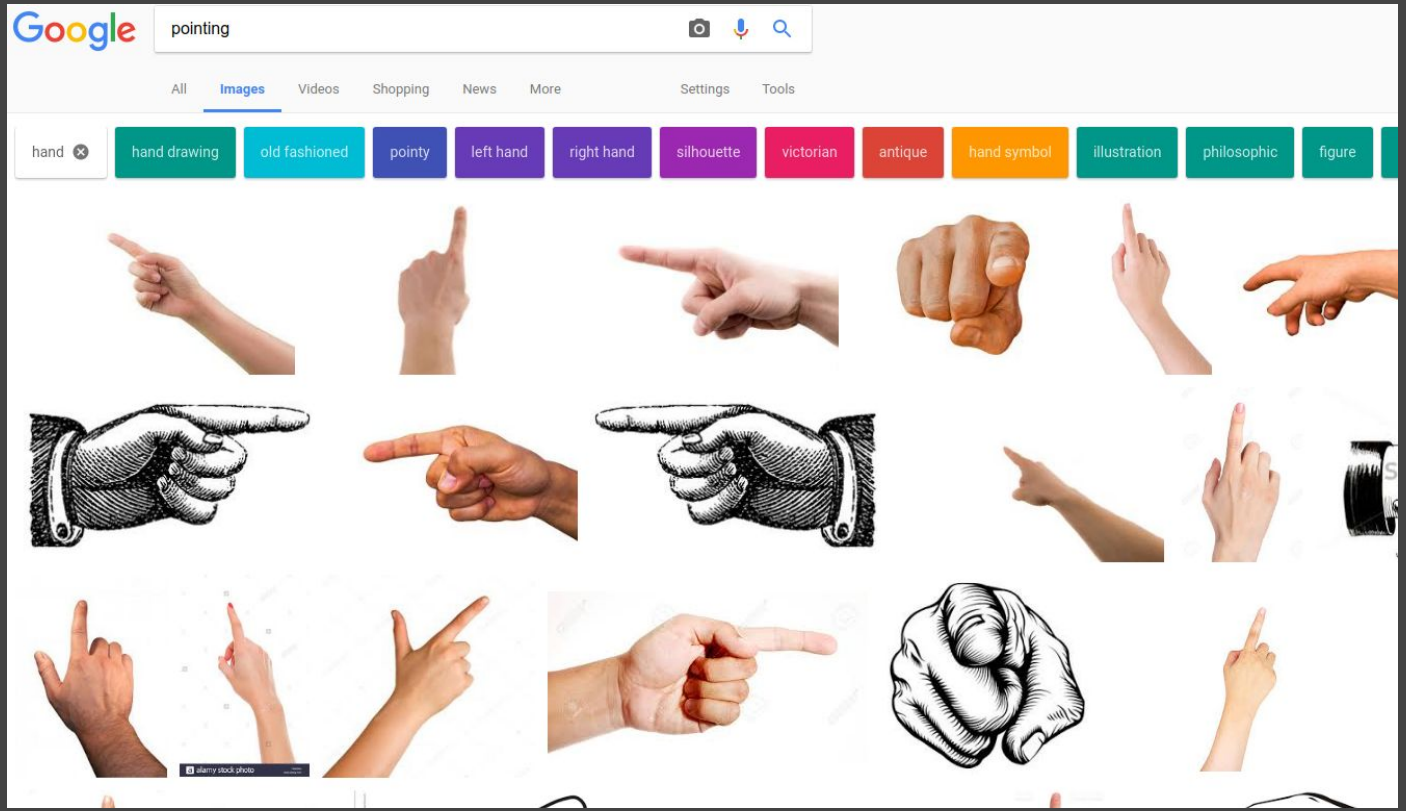
What is a Pointer?



What is a Pointer?

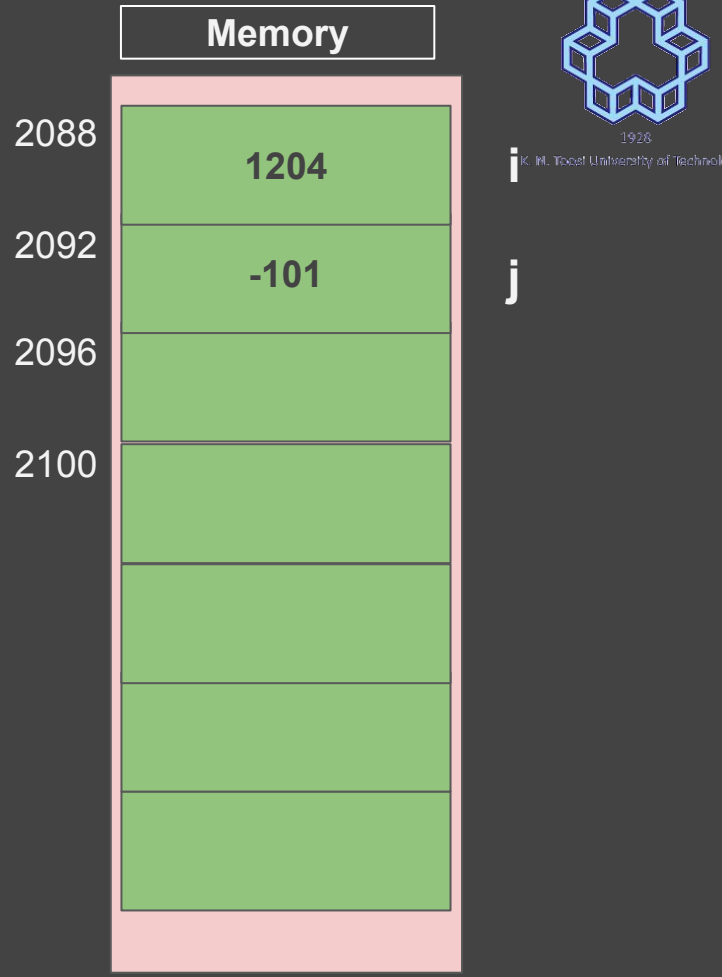


What is a Pointer?



What is a Pointer?

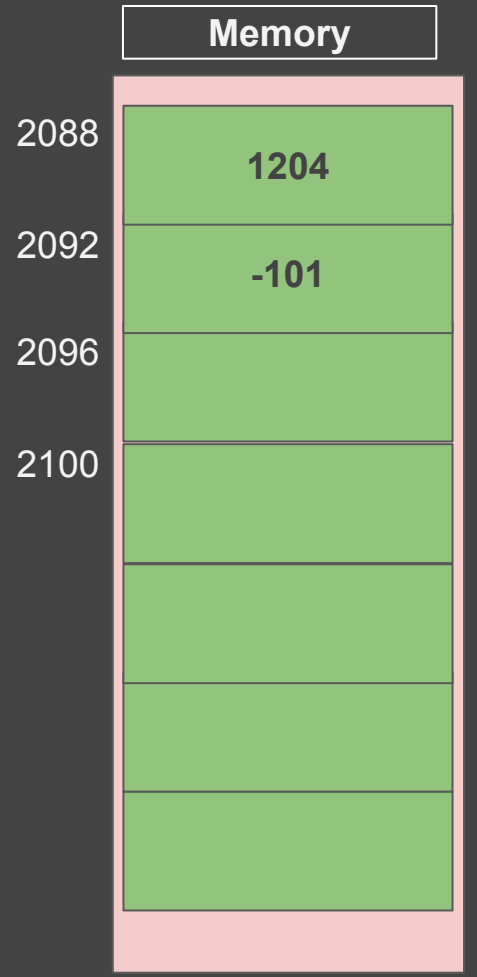
```
int i = 1204;
int j = -101;
```



What is a Pointer?

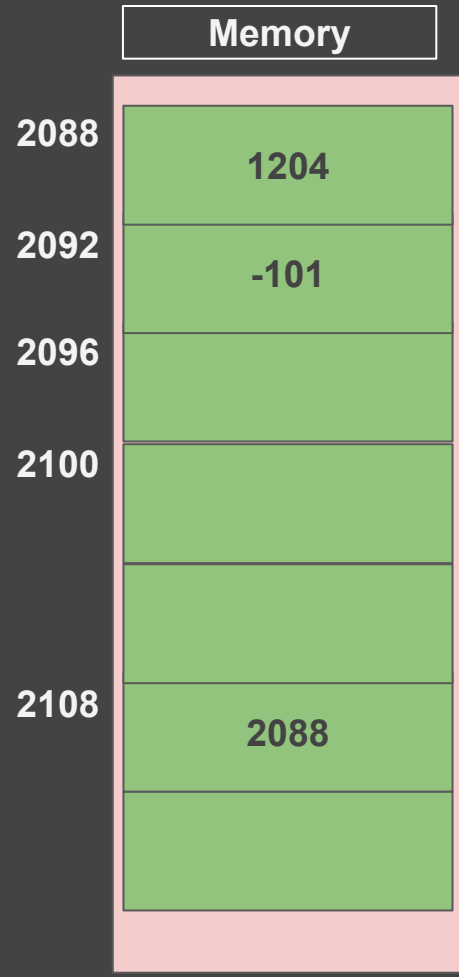
```
int i = 1204;
int j = -101;
printf("%u\n", &i);
```

i	j	&i	&j
1204	-101	2088	2092



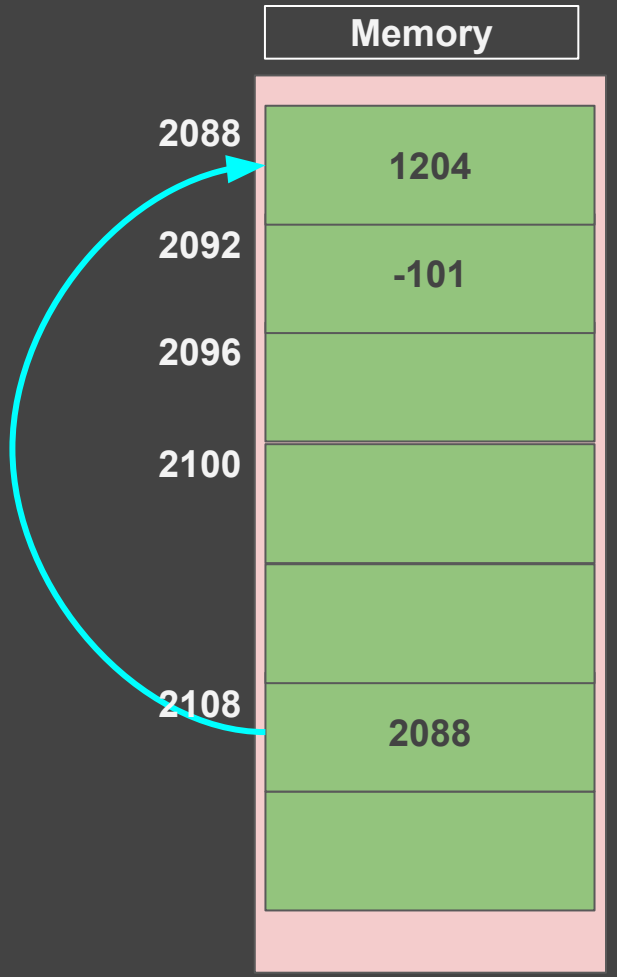
What is a Pointer?

```
int i = 1204;
int j = -101;
```



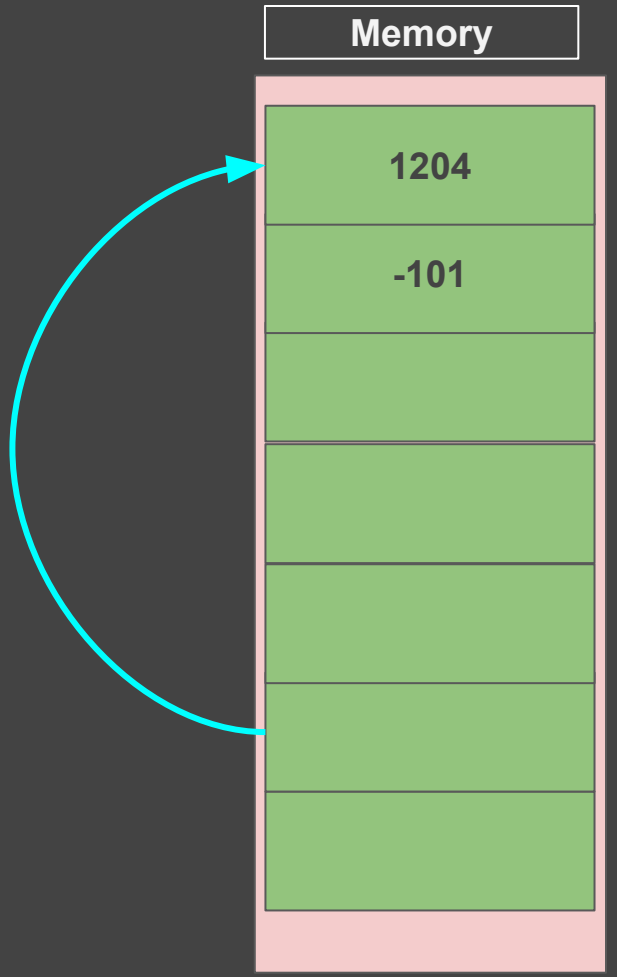
What is a Pointer?

```
int i = 1204;
int j = -101;
```



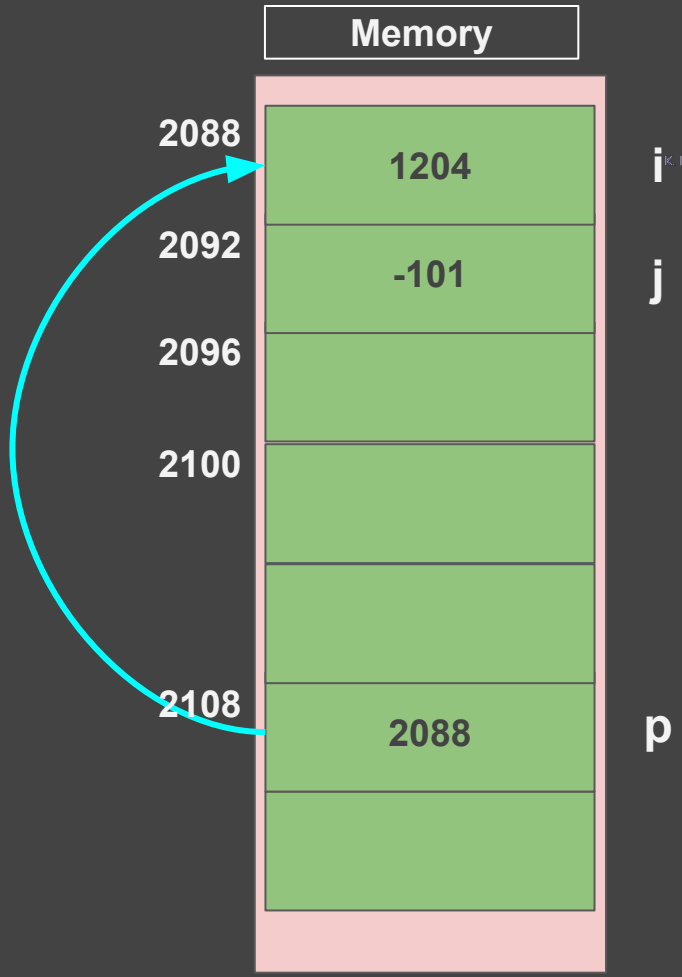
What is a Pointer?

```
int i = 1204;
int j = -101;
```



What is a Pointer?

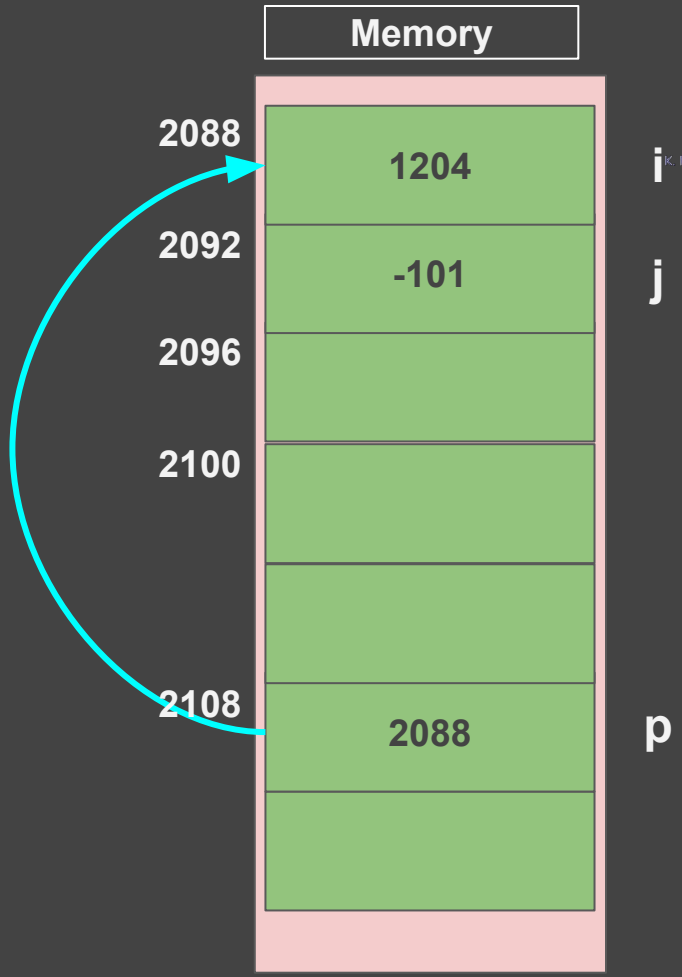
```
int i = 1204;
int j = -101;
int *p;

p = &i;
```



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

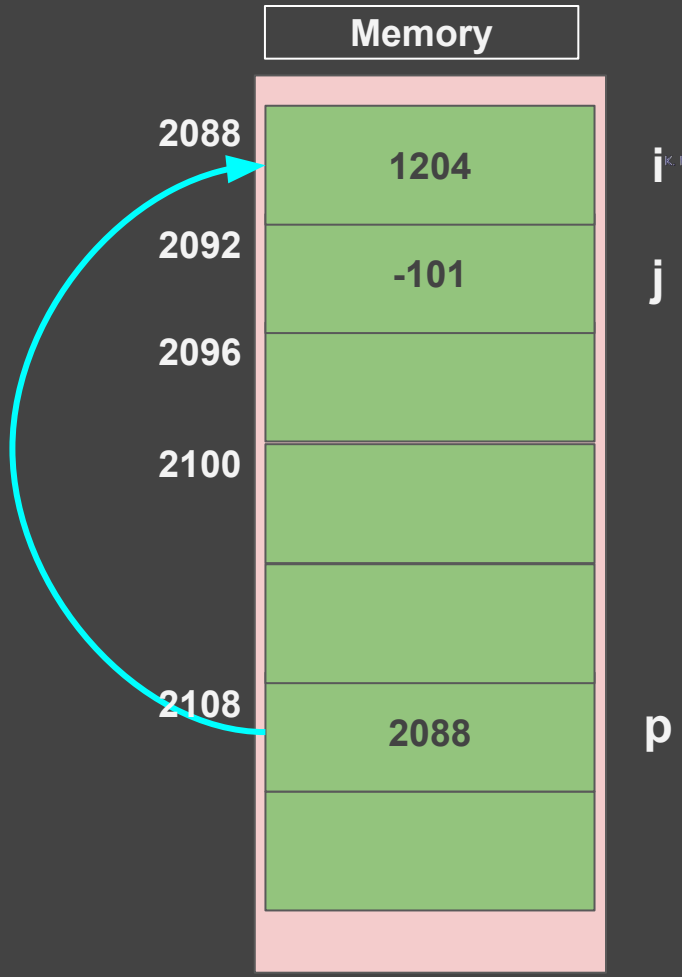
i	p	&i	&p
?	?	?	?



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

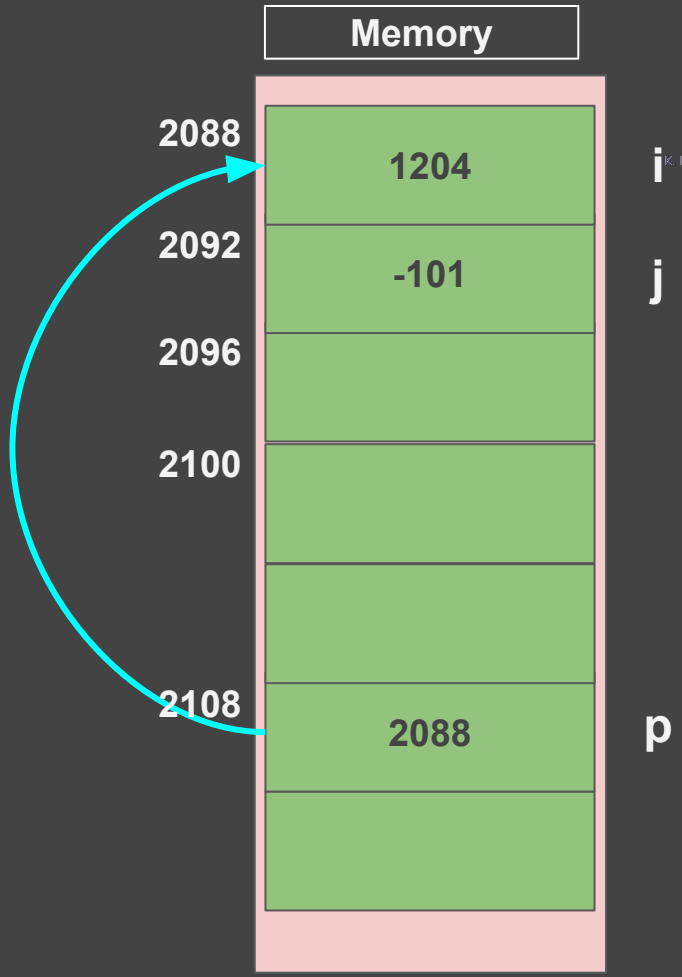
i	p	&i	&p
1204	?	?	?



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

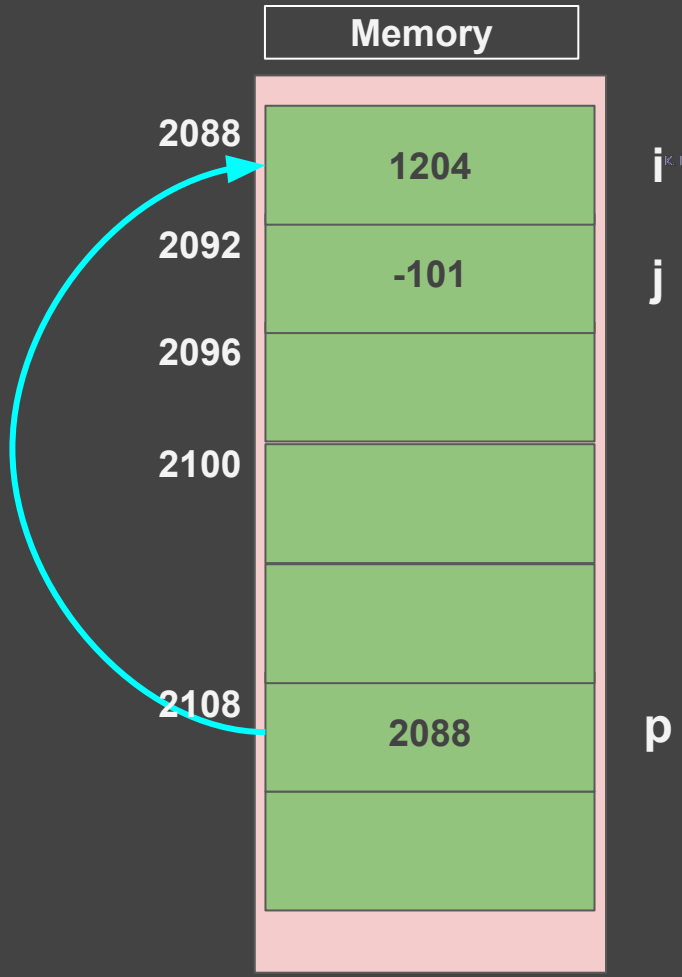
i	p	&i	&p
1204	2088	?	?



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

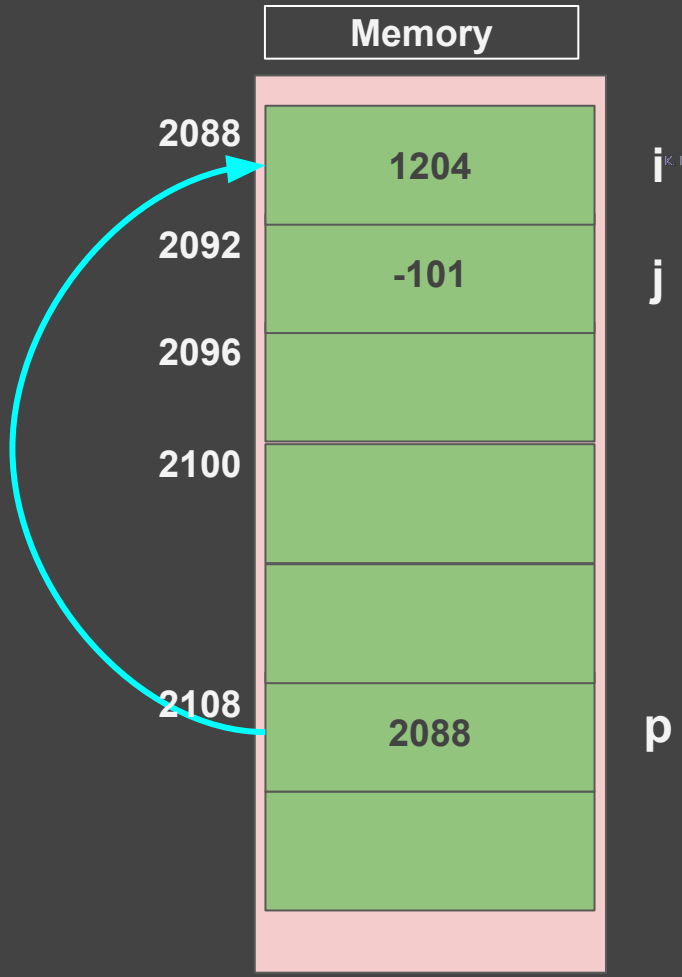
i	p	&i	&p
1204	2088	2088	?



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

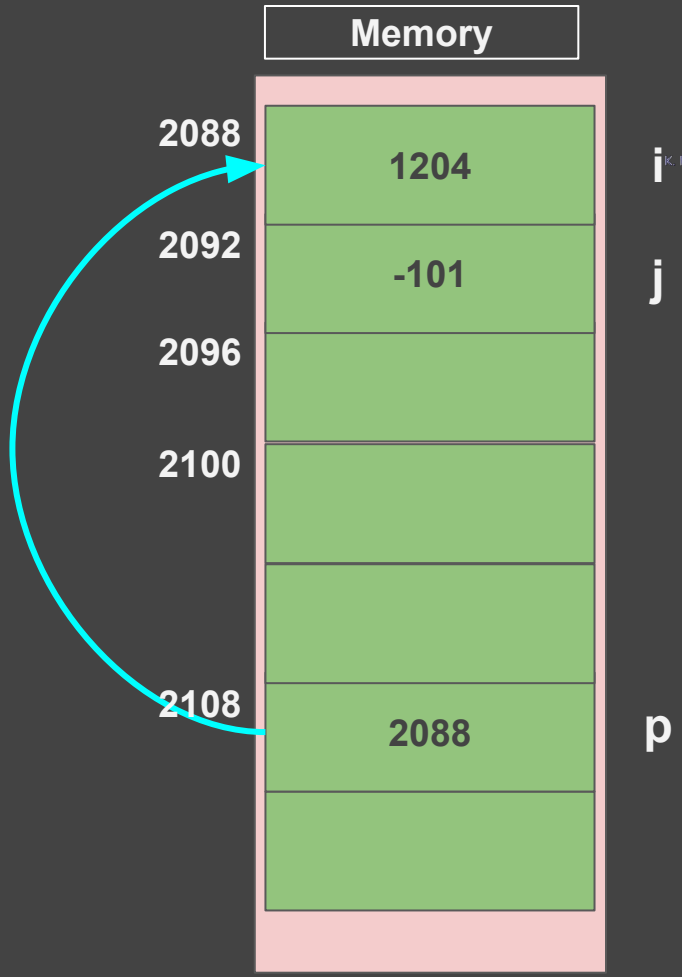
i	p	&i	&p
1204	2088	2088	2108



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;

p = &i;
```

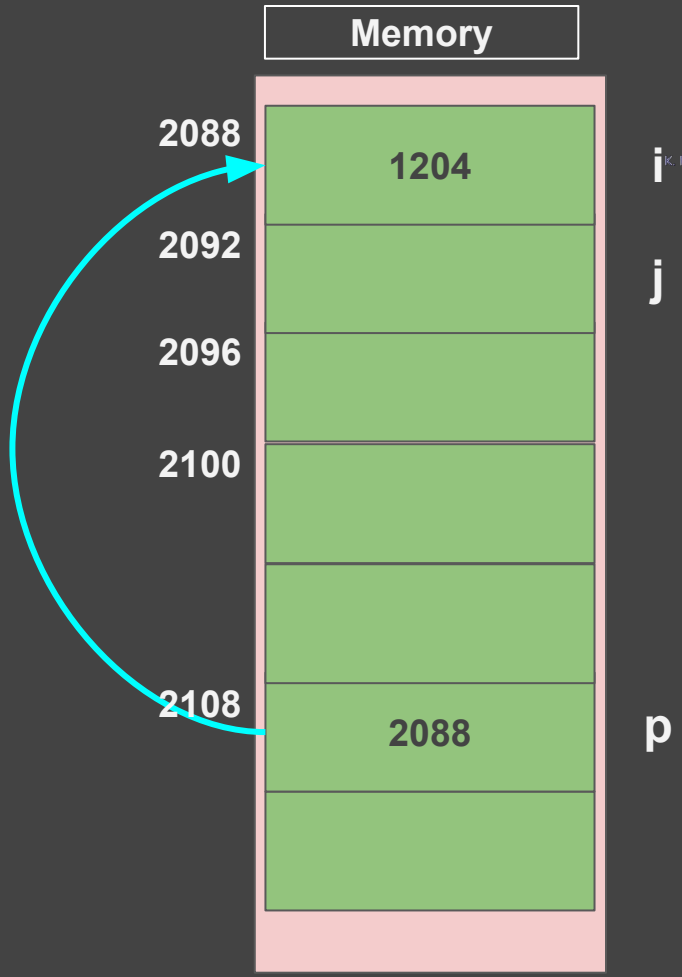
i	p	&i	&p
1204	2088	2088	2108



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;
p = &i;
j = *p;
```

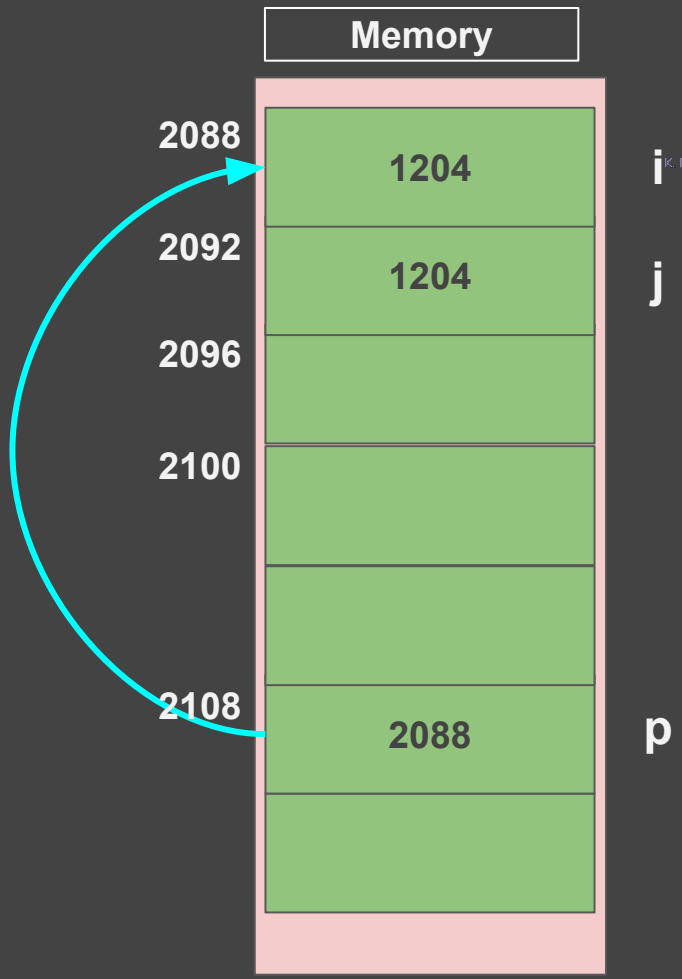
i	p	&i	&p	*p
1204	2088	2088	2108	?



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;
p = &i;
j = *p;
```

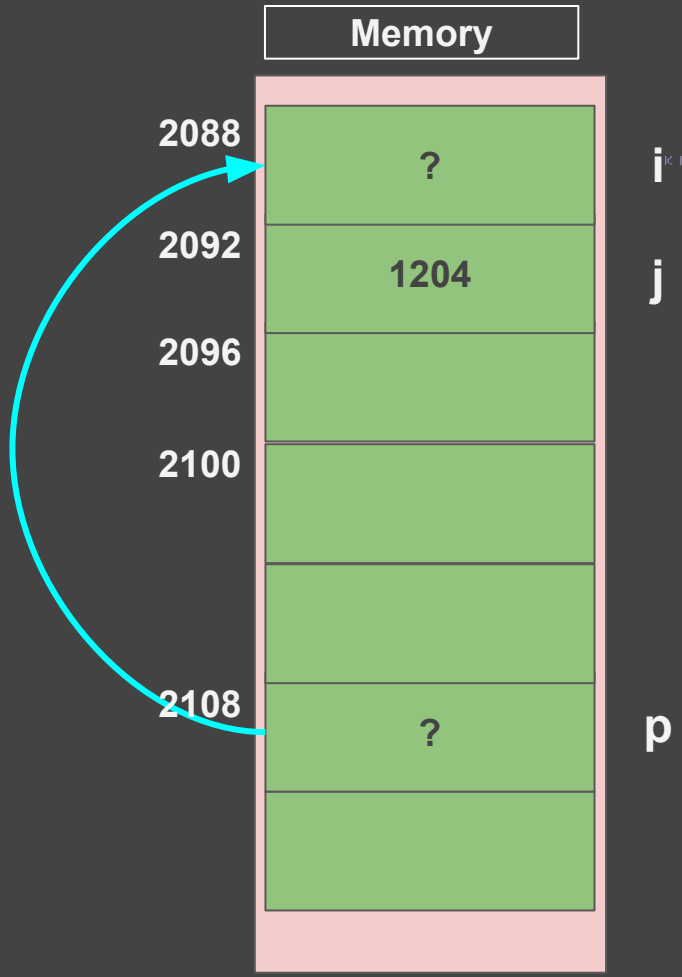
i	p	&i	&p	*p
1204	2088	2088	2108	1204



What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;
p = &i;
j = *p;
*p = -22;
```

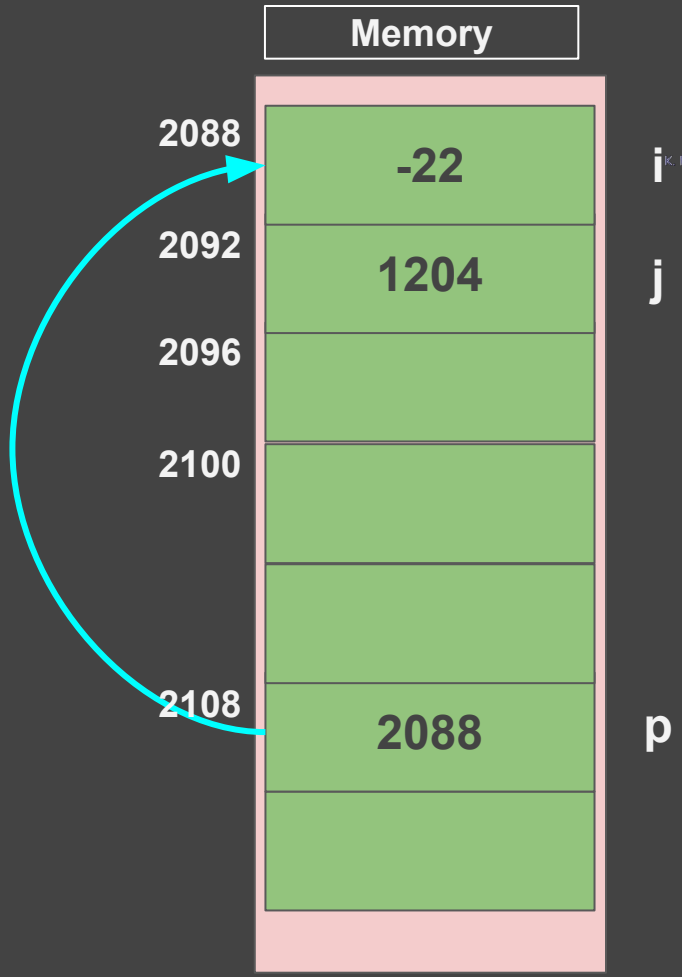
i	p	&i	&p	*p
?	?	?	?	?



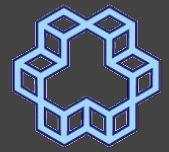
What is a Pointer?

```
int i = 1204;
int j = -101;
int *p;
p = &i;
j = *p;
*p = -22;
```

i	p	&i	&p	*p
-22	2088	2088	2108	-22

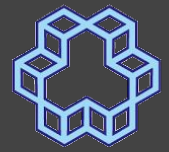


Simulating by-reference argument passing




```c
#include <stdio.h>

void duplicate(int x);

int main() {
  int x = 10;

  duplicate(x);
  printf("%d\n",x);

  return 0;
}

void duplicate(int x) {
  x *= 0;
}
```

Simulating by-reference argument passing

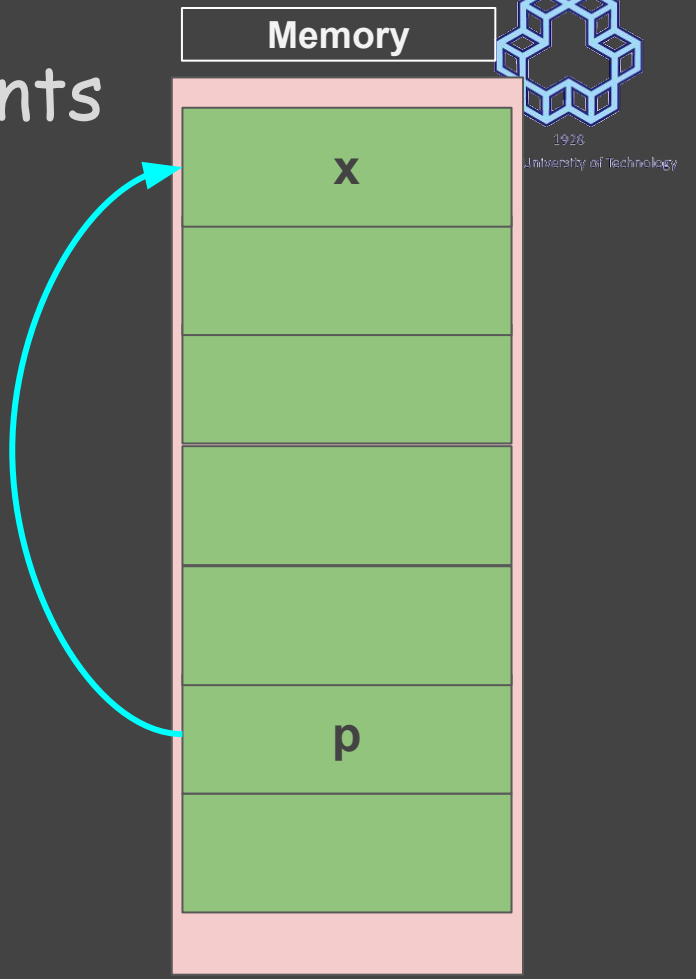



```c
#include <stdio.h>

void duplicate(int x);

int main() {
  int x = 10;

  duplicate(x);
  printf("%d\n",x);

  return 0;
}

void duplicate(int x) {
  x *= 0;
}
```

```c
#include <stdio.h>

void duplicate(int*);

int main() {
  int x = 10;

  duplicate(&x);
  printf("%d\n",x);

  return 0;
}

void duplicate(int *p) {
  *p *= 2;
}
```

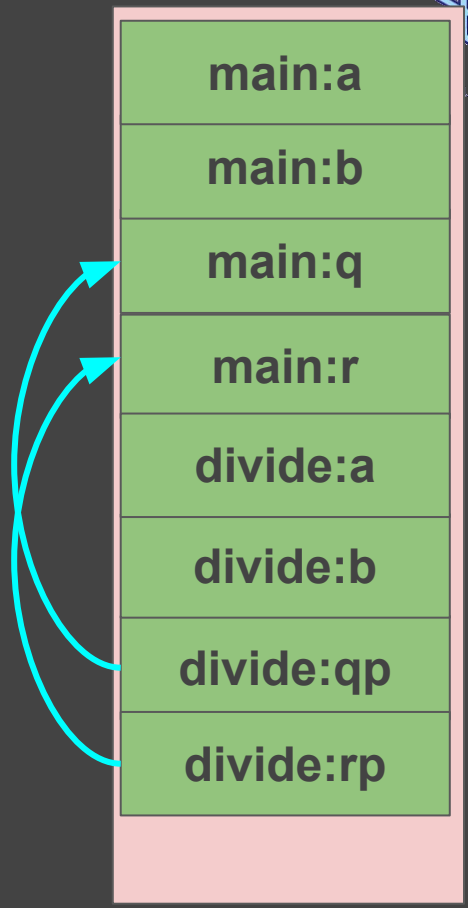
Simulating by-reference arguments

```c
#include <stdio.h>

void duplicate(int*);

int main() {
  int x = 10;

  duplicate(&x);
  printf("%d\n",x);

  return 0;
}

void duplicate(int *p) {
  *p *= 2;
}
```



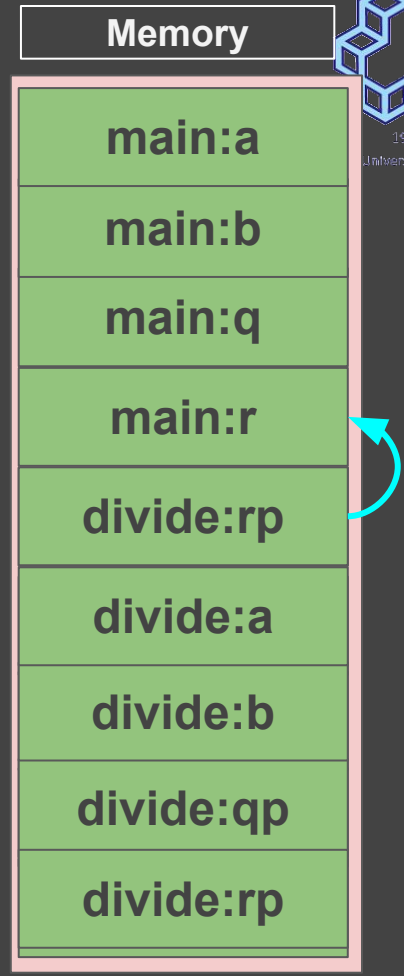
multiple outputs

```c
#include <stdio.h>

void divide(int a, int b, int *qp, int *rp);

int main() {
  int a,b,q,r;

  scanf("%d %d", &a, &b);

  divide(a,b,&q,&r);

  printf("%d = %d * %d + %d\n", a, q, b, r);

  return 0;
}

void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
}
```



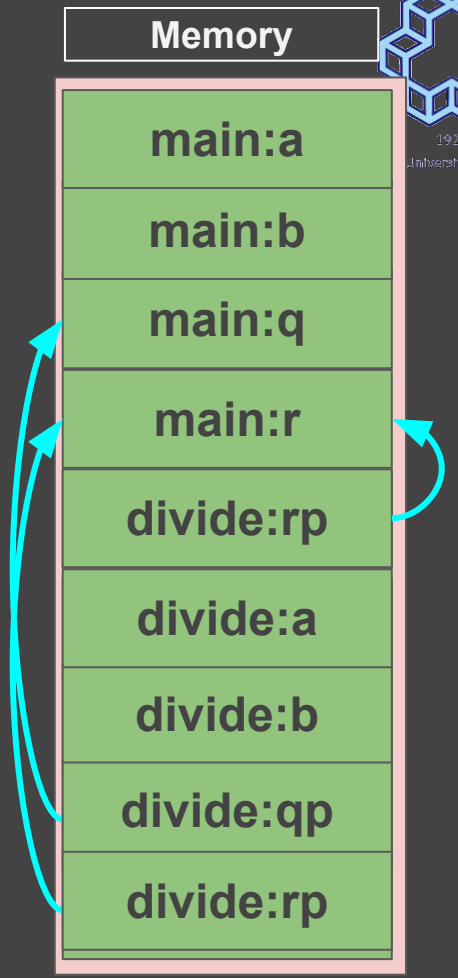
Arguments are still by-value

```c
int main() {
  int a,b,q,r;
  int *rp;

  scanf("%d %d", &a, &b);

  rp = &r;
  divide(a,b,&q,rp);

  printf("%d = %d * %d + %d\n", a, q, b, r);
  printf("rp = %lu\n", rp);

  return 0;
}

void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
    rp = &a;

    printf("rp = %lu\n", rp);
}
```



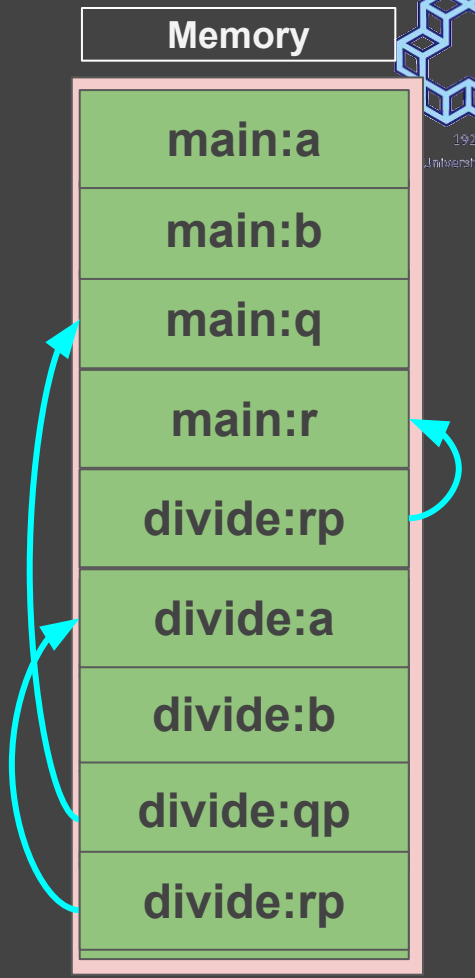
Arguments are still by-value

```c
int main() {
  int a,b,q,r;
  int *rp;

  scanf("%d %d", &a, &b);

  rp = &r;
  divide(a,b,&q,rp);

  printf("%d = %d * %d + %d\n", a, q, b, r);
  printf("rp = %lu\n", rp);

  return 0;
}

void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
    rp = &a;

    printf("rp = %lu\n", rp);
}
```



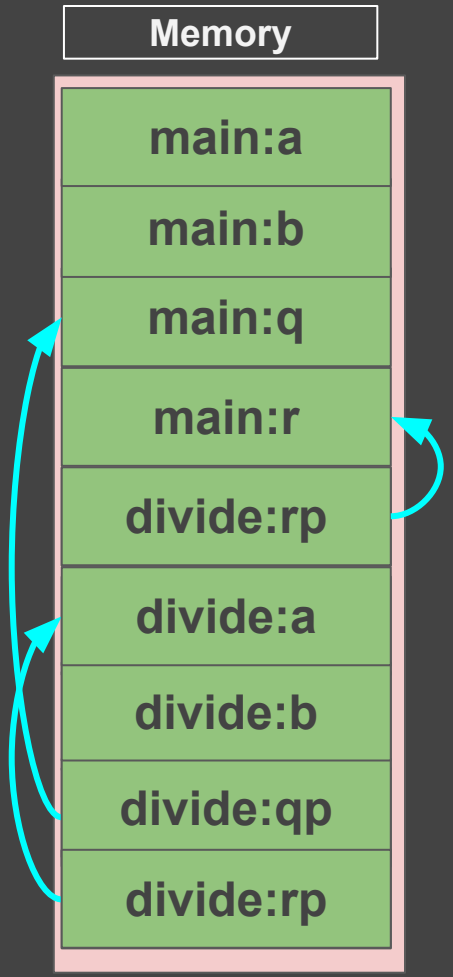
Arguments are still by-value

```c
int main() {
  int a,b,q,r;
  int *rp;

  scanf("%d %d", &a, &b);

  rp = &r;
  divide(a,b,&q,rp);

  printf("%d = %d * %d + %d\n", a, q, b, r);
  printf("rp = %lu\n", rp);

  return 0;
}

void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
    rp = &a;

    printf("rp = %lu\n", rp);
}
```

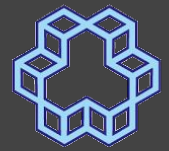


use %p to print addresses

```c
int main() {
  int a,b,q,r;
  int *rp;

  scanf("%d %d", &a, &b);

  rp = &r;
  divide(a,b,&q,rp);

  printf("%d = %d * %d + %d\n", a, q, b, r);
  printf("rp = %p\n", rp);

  return 0;
}

void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
    rp = &a;

    printf("rp = %p\n", rp);
}
```



Example: Swapping

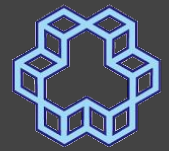



```c
#include <stdio.h>

void swap(double *p, double *q);

int main() {
  double x,y;

  scanf("%lf %lf", &x, &y);

  printf("x=%.1f, y=%.1f\n", x, y);

  swap(&x,&y);

  printf("x=%.1f, y=%.1f\n", x, y);

  return 0;
}
```

```c
void swap(double *p, double *q) {
  double temp;
  temp = *p;
  *p = *q;
  *q = temp;
}
```

pointerfunc5_swap.c

constant pointers



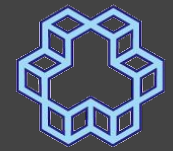

```c
#include <stdio.h>


int main() {
  int a=0, b=1;

  int *p = &a;

  p = &b;

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

}
```

constant pointers


```c
#include <stdio.h>


int main() {
  int a=0, b=1;

  int *p = &a;

  p = &b;

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

}
```

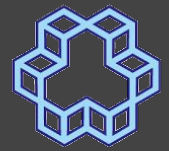
```
nasihatkon@kntu:code$ gcc pointerconst1.c && ./a.out
1
nasihatkon@kntu:code$
```

constant pointers




```c
#include <stdio.h>


int main() {
  int a=0, b=1;

  const int *p = &a;

  p = &b;

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

}
```

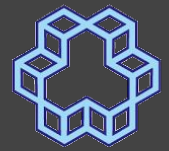
constant pointers




```c
#include <stdio.h>


int main() {
  int a=0, b=1;

  const int *p = &a;

  p = &b;

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

}
```

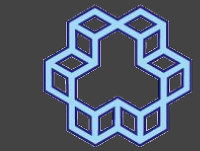
```
nasihatkon@kntu:code$ gcc pointerconst2.c && ./a.out
1
nasihatkon@kntu:code$
```

constant pointers




```c
#include <stdio.h>

int main() {
  int a=0, b=1;

  const int *p = &a;

  *p = 2;

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

}
```

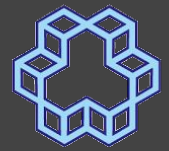
constant pointers




```c
#include <stdio.h>

void #include <stdio.h>

int rint main() {
  int   int a=0, b=1;

  cor   const int *p = &a;

  *p    *p = 2;

  pri   printf("%d\n", *p);

}     }
```

```
nasihatkon@kntu:code$ gcc pointerconst3.c && ./a.out
pointerconst3.c: In function ‘main’:
pointerconst3.c:12:6: error: assignment of read-only location ‘*p’
   *p = 2;
      ^
nasihatkon@kntu:code$
```

constant pointers



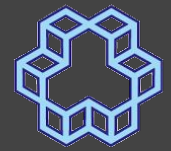

```c
#include <stdio.h>

int main() {
  int a=0, b=1;

  int * const p = &a;

  *p = 2;

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

}
```

constant pointers


```c
#include <stdio.h>

int main() {
  int a=0, b=1;

  int * const p = &a;

  *p = 2;

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

}
```

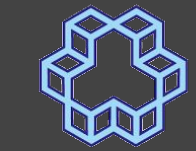
```
nasihatkon@kntu:code$ gcc pointerconst4.c && ./a.out
2
nasihatkon@kntu:code$
```

constant pointers




```c
#include <stdio.h>

void swap(double *, double *);

int main() {
  int a=0, b=1;

  int * const p = &a;

  p = &b;

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

}
```

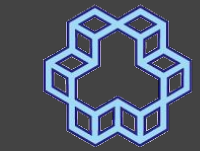
constant pointers




```c
#include <stdio.h>

void swap(double *, double *);

int main() {
  int a=0, b=1;

  int * const p = &a;

  p = &b;

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

}
```

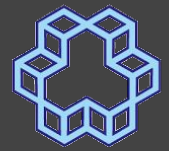
```
nasihatkon@kntu:code$ gcc pointerconst5.c && ./a.out
pointerconst5.c: In function ‘main’:
pointerconst5.c:10:5: error: assignment of read-only variable ‘p’
   p = &b;
     ^
```

constant pointers




```c
#include <stdio.h>

int main() {
  int a=0, b=1;

  const int * const p = &a;

  *p = 2;

  p = &b;


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

}
```

constant pointers



same thing

```
const int   *p1 = &a;
int const   *p2 = &a;
int * const p3 = &a;
```