



Fundamentals of Programming

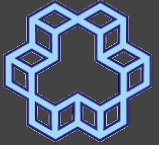
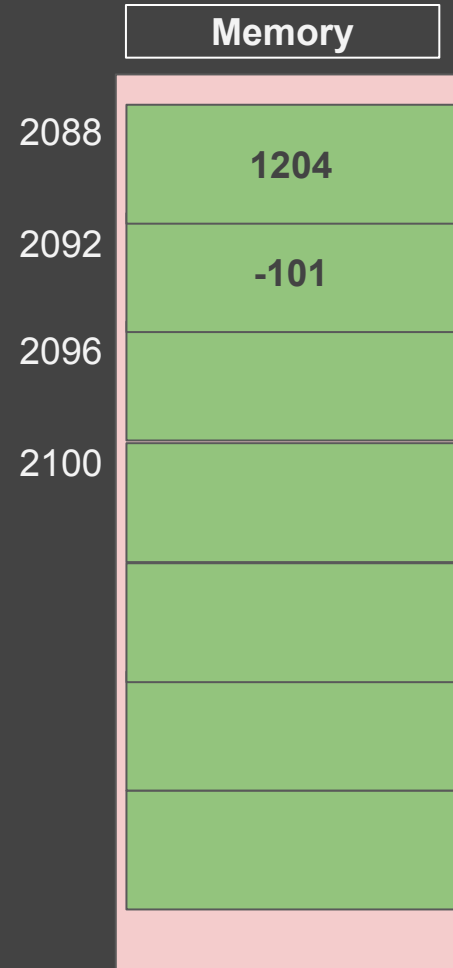
lecture 18

Introduction to Pointers

```
*****  
* convolve.c  
***** /  
  
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdlib.h> /* malloc(), realloc() */  
  
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */  
  
#define MAX_KERNEL_WIDTH 71  
  
typedef struct {  
    int ndim;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;  
  
/* Kernels */
```

Variable address

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

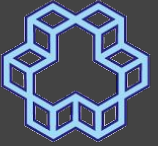


1926

K. J. Somaiya Institute of Technology

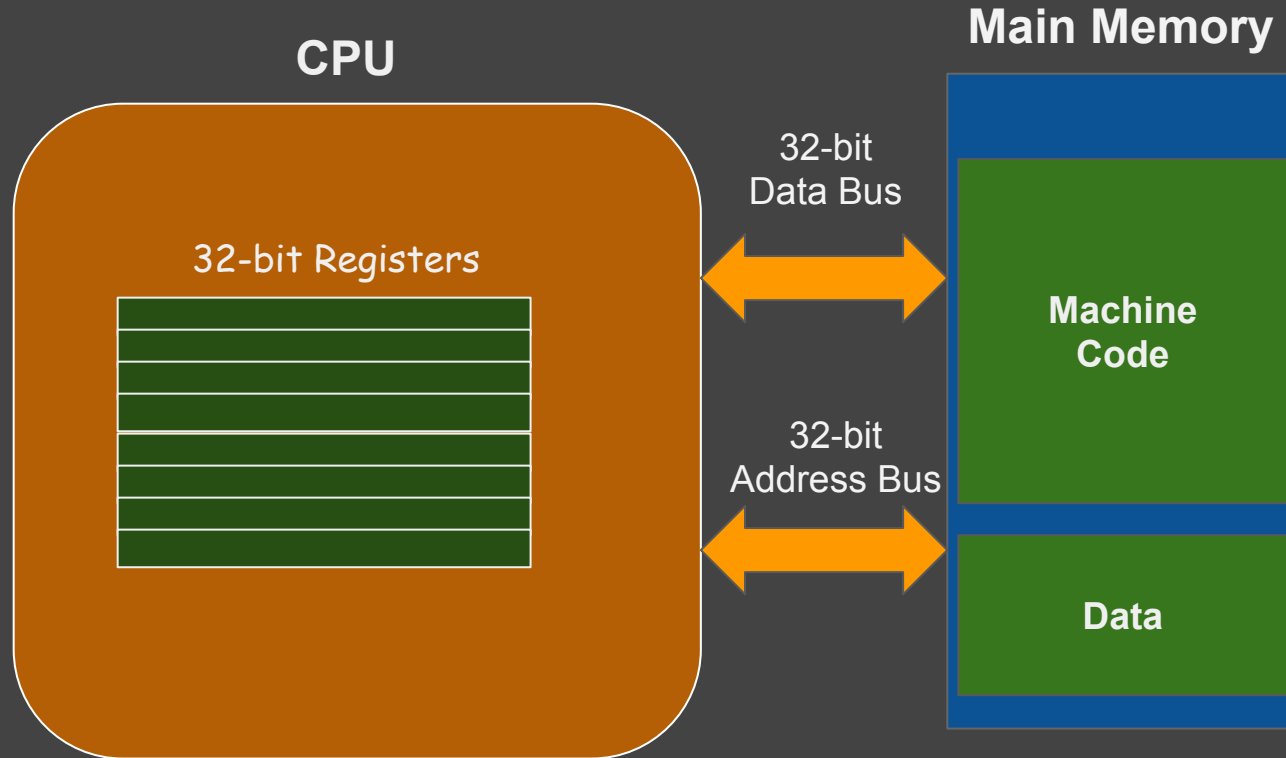
j

32-bit system

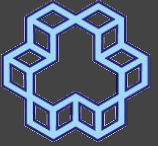


1926

K. N. Toosi University of Technology

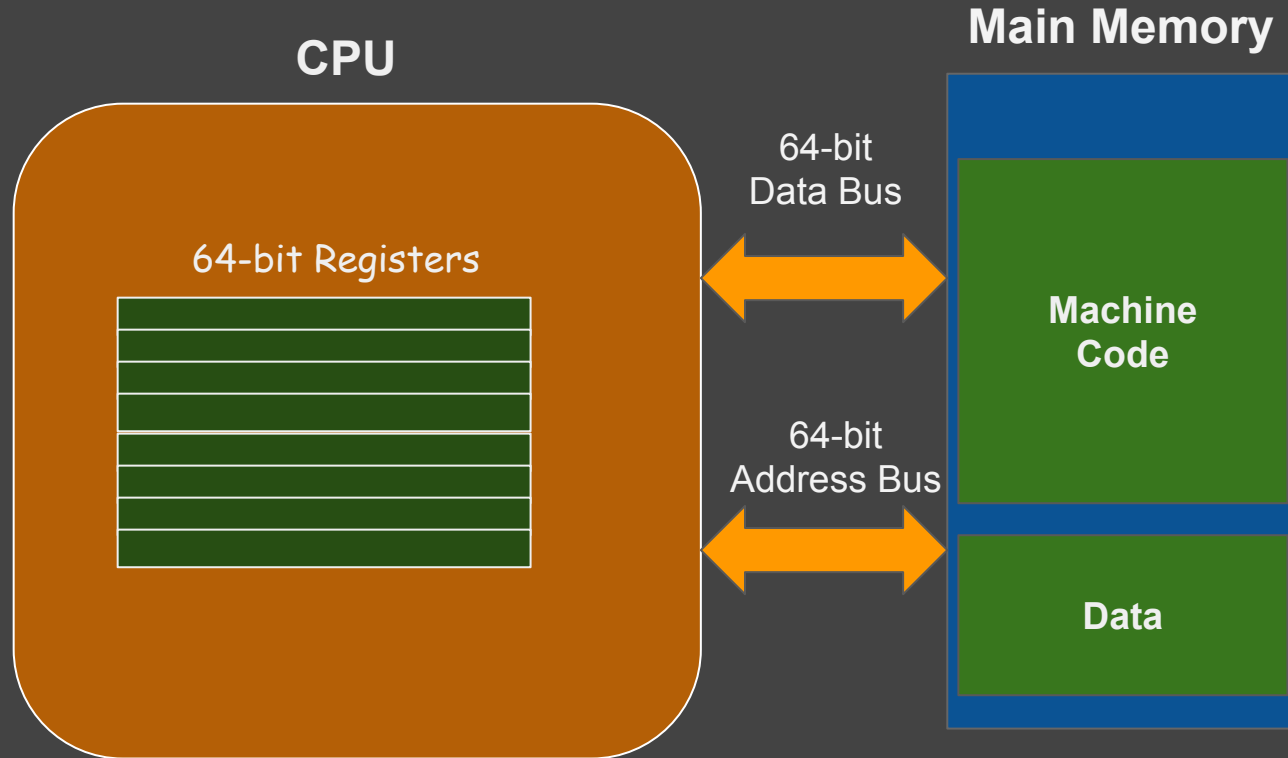


64-bit system



1926

K. N. Toosi University of Technology



Variable address

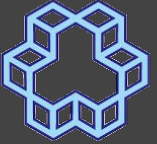
```
int i = 1204;  
  
int j = -101;  
  
printf("%d %d \n", i, j);
```

address1.c

Memory

1204

-101



1926

K. J. Somaiya Institute of Technology

j

Variable address

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

address2.c

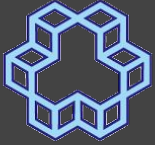
Memory

1204

-101

i

j



1926

K. N. Toosi University of Technology

Variable address

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

address2.c

```
1204 -101  
140733182053744 140733182053748
```

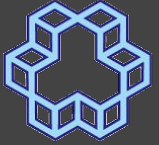
Memory

1204

-101

i

j



1926

K. N. Toosi University of Technology

Variable address

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

address2.c

```
1204 -101
140733182053744 140733182053748
```

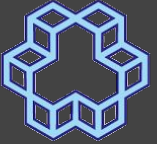
140733182053744

140733182053748

Memory

1204

-101



1926

K. N. Toosi University of Technology

j

Variable address

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

address3.c

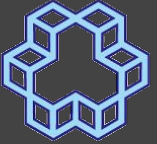
140733182053744

140733182053748

Memory

1204

-101



1926

K. J. Somaiya Institute of Technology

j

Variable address

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

address3.c

```
1204 -101  
7ffeb6925c20 7ffeb6925c24
```

7ffeb6925c20

7ffeb6925c24

Memory

1204

-101

i

j



1926

K. N. Toosi University of Technology

Variable address

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

address4.c

7ffeb6925c20

7ffeb6925c24

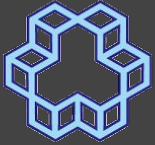
Memory

1204

-101

i

j



1926

K. N. Toosi University of Technology

Variable address

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

address4.c

```
lecture18$ gcc address4.c && ./a.out
```

```
1204 -101
```

```
0x7ffeca6ebcc0 0x7ffeca6ebcc4
```

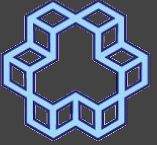
7ffeb6925c20

7ffeb6925c24

Memory

1204

-101



1926

K. N. Toft University of Technology

j

Variable address

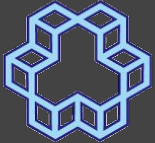
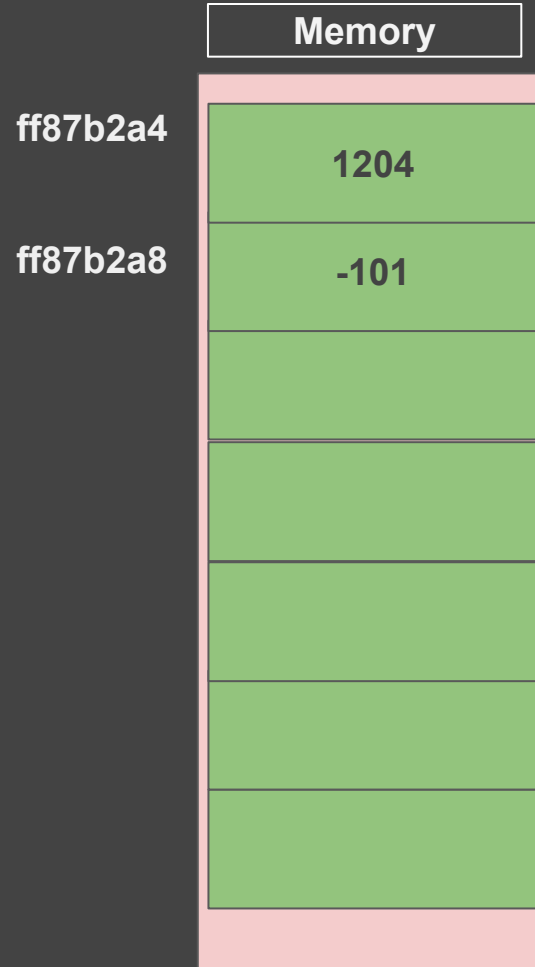
```
int i = 1204;  
  
int j = -101;  
  
printf("%d %d \n", i, j);  
  
printf("%p %p \n", &i, &j);
```

address4.c

```
lecture18$ gcc -m32 address4.c && ./a.out
```

```
1204 -101
```

```
0xff87b2a4 0xff87b2a8
```



1926

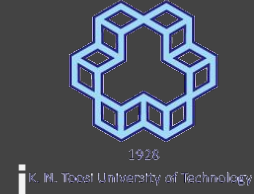
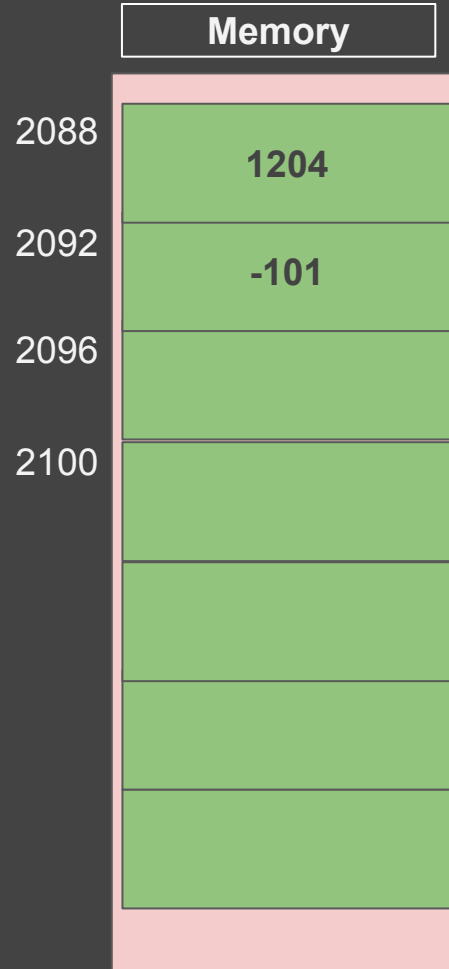
K. N. Toosi University of Technology

j

Variable address

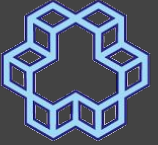
```
int i = 1204;  
int j = -101;  
printf("%u\n", &i);
```

i	j	&i	&j
1204	-101	2088	2092



j

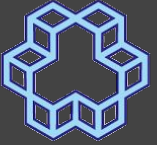
What is a Pointer?



1926

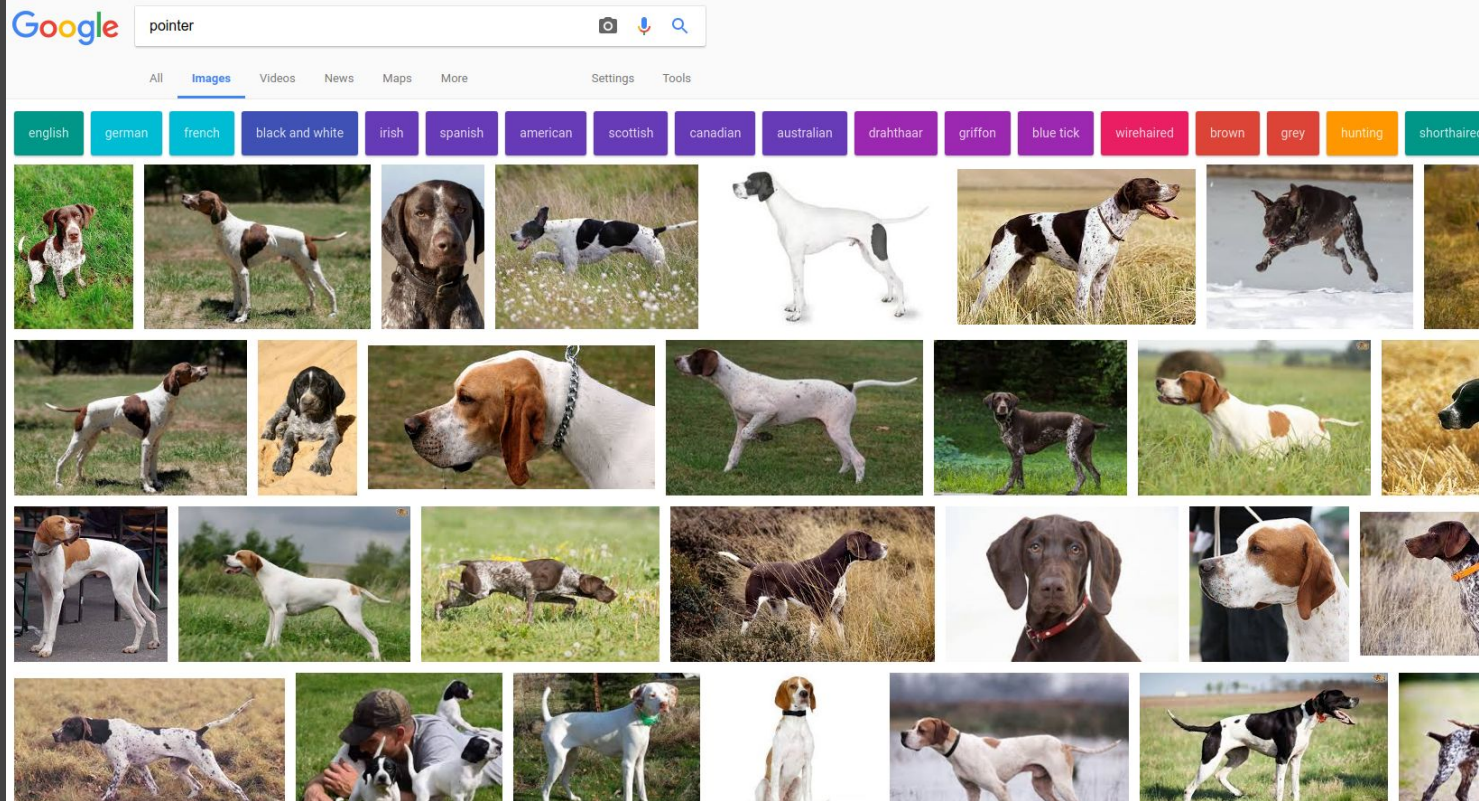
K. J. Somaiya Institute of Technology

What is a Pointer?



1926

K. J. Somaiya Institute of Technology

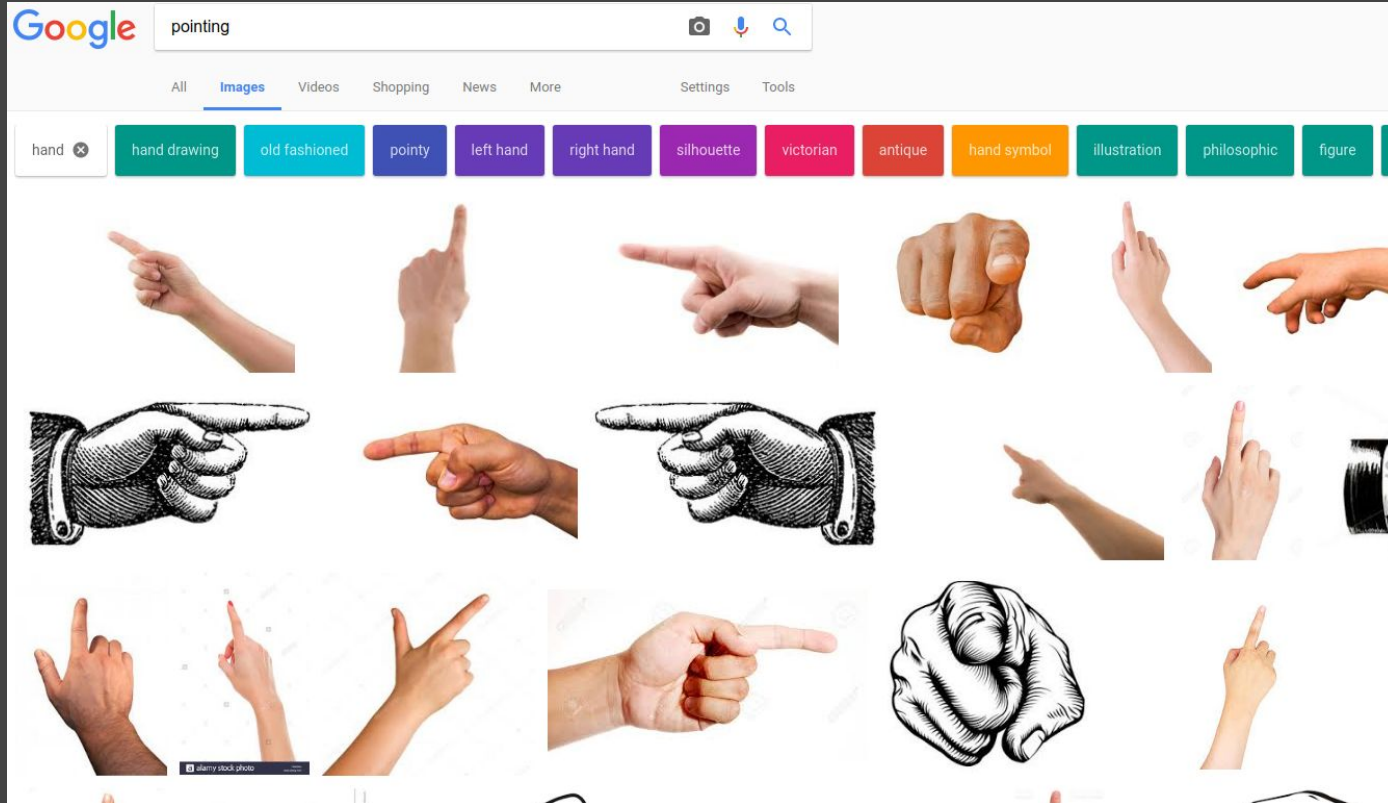


What is a Pointer?



1926

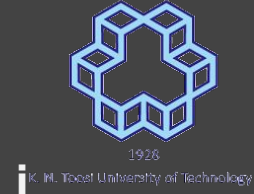
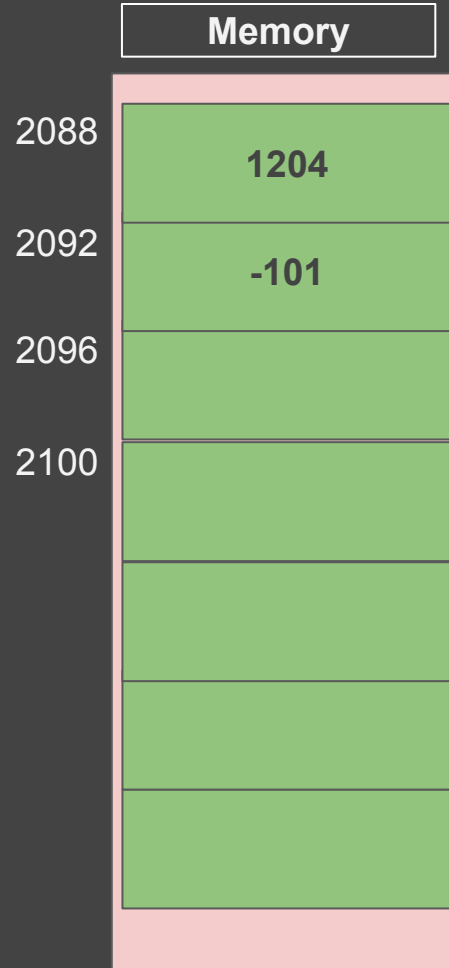
K. N. Toosi University of Technology



What is a Pointer?

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

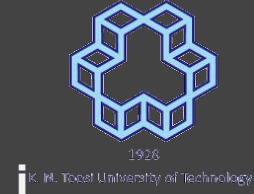
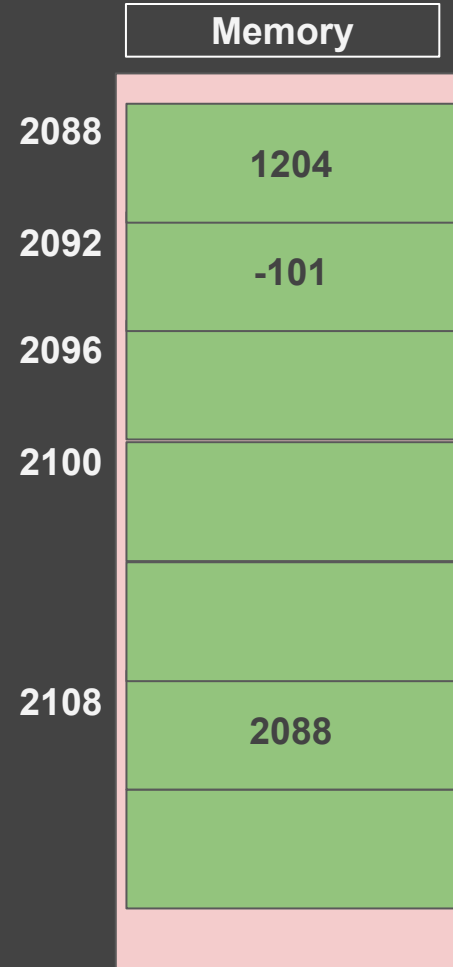
i	j	&i	&j
1204	-101	2088	2092



j

What is a Pointer?

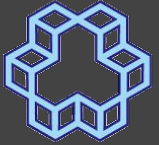
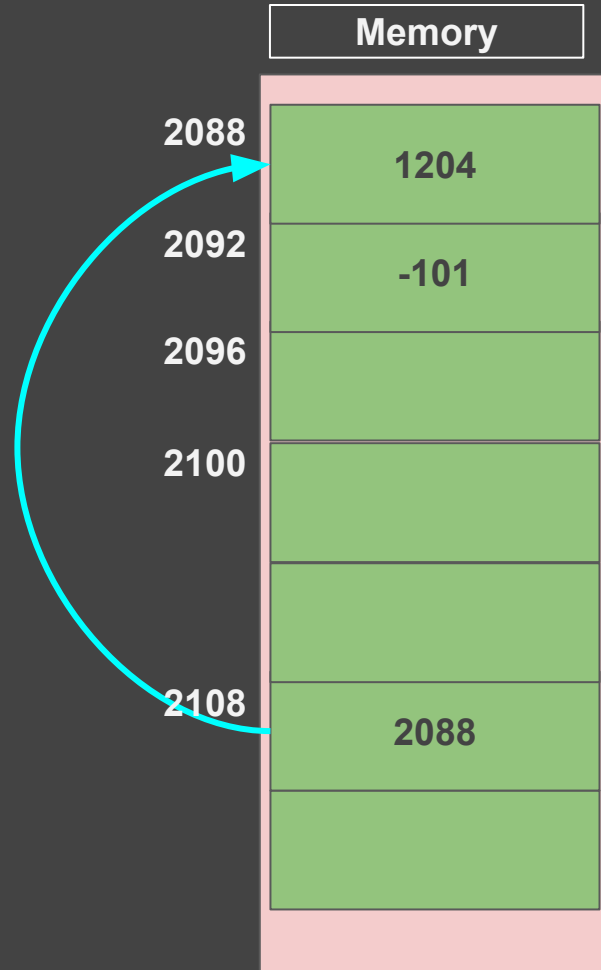
```
int i = 1204;  
int j = -101;
```



j

What is a Pointer?

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



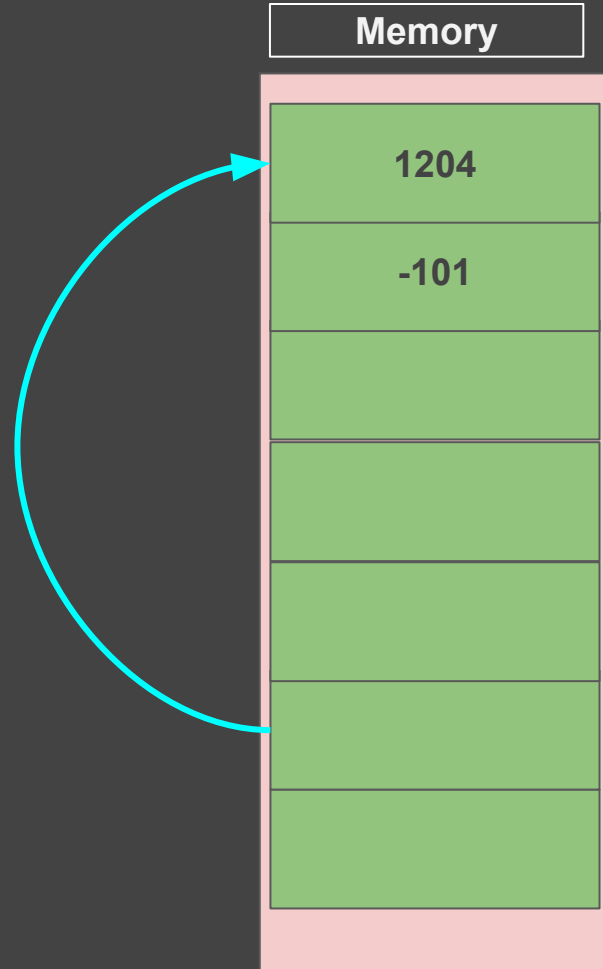
1926

K. J. Somaiya Institute of Technology

j

What is a Pointer?

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



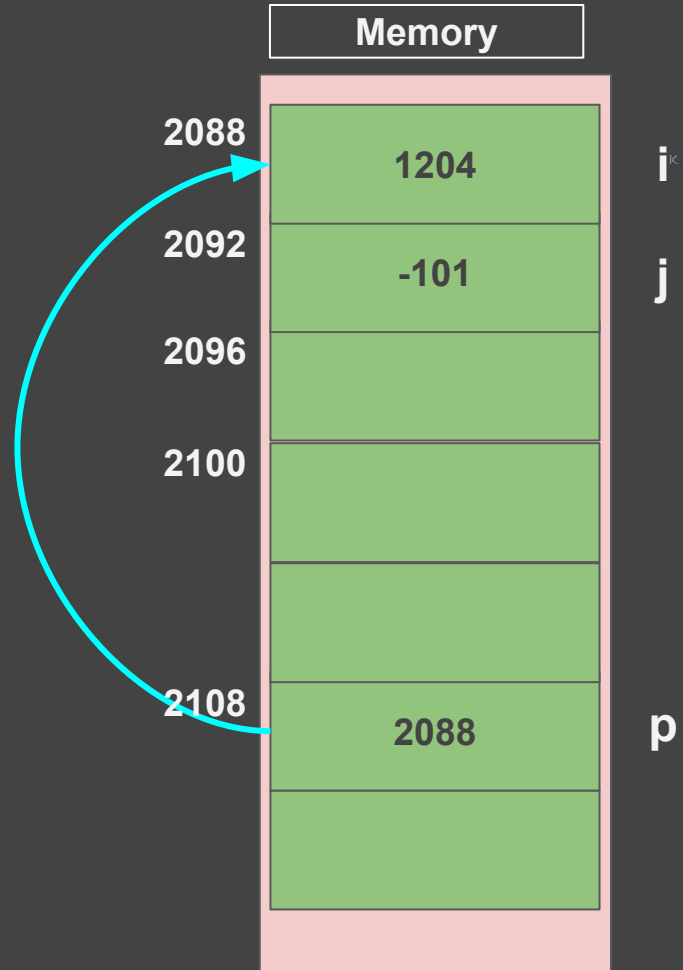
1926

K. J. Somaiya Institute of Technology

j

What is a Pointer?

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



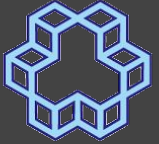
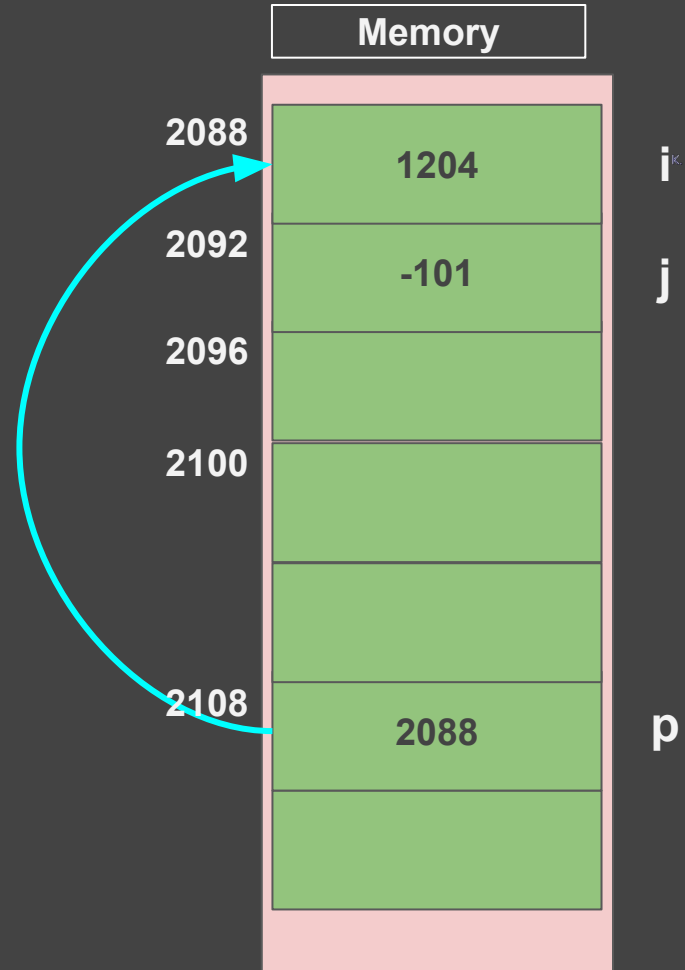
1926

K. J. Somaiya Institute of Technology

What is a Pointer?

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

i	p	&i	&p
?	?	?	?



1926

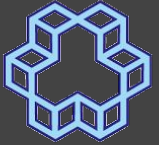
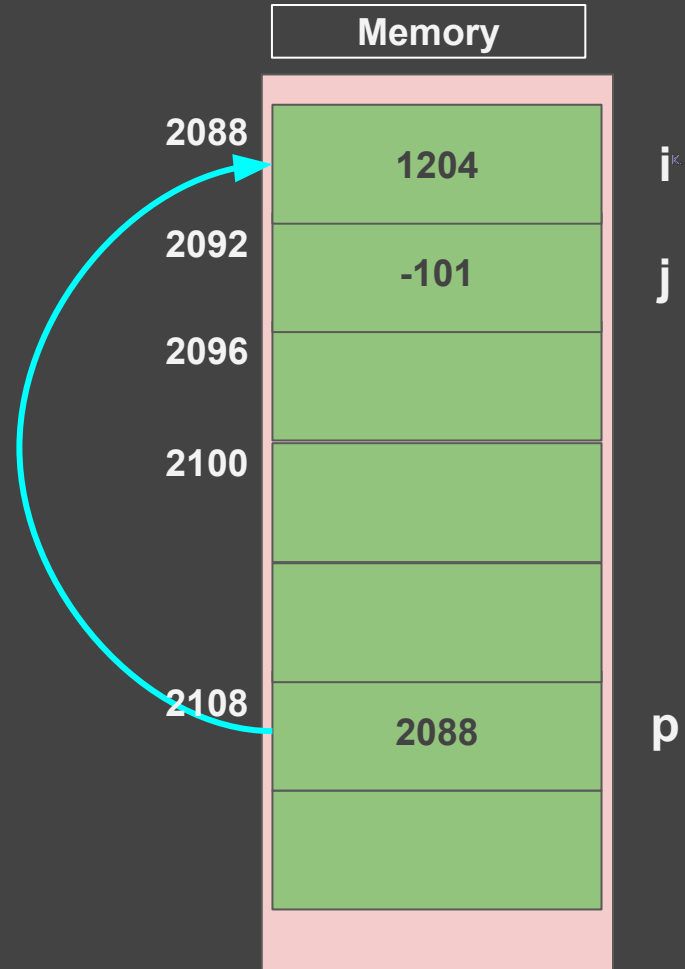
K. J. Somaiya Institute of Technology

What is a Pointer?

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

```
p = &i;
```

i	p	&i	&p
1204	?	?	?



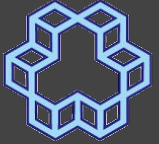
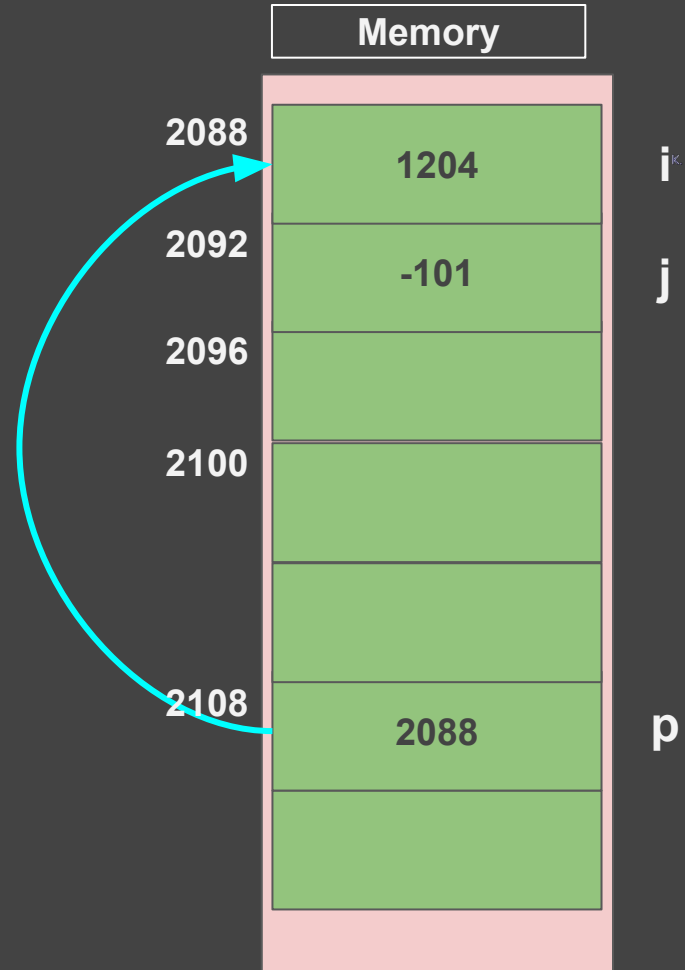
1926

K. J. Somaiya Institute of Technology

What is a Pointer?

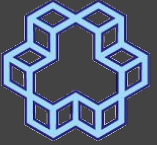
```
int i = 1204;  
int j = -101;  
int *p;  
  
p = &i;
```

i	p	&i	&p
1204	2088	?	?



1926

K. J. Somaiya Institute of Technology



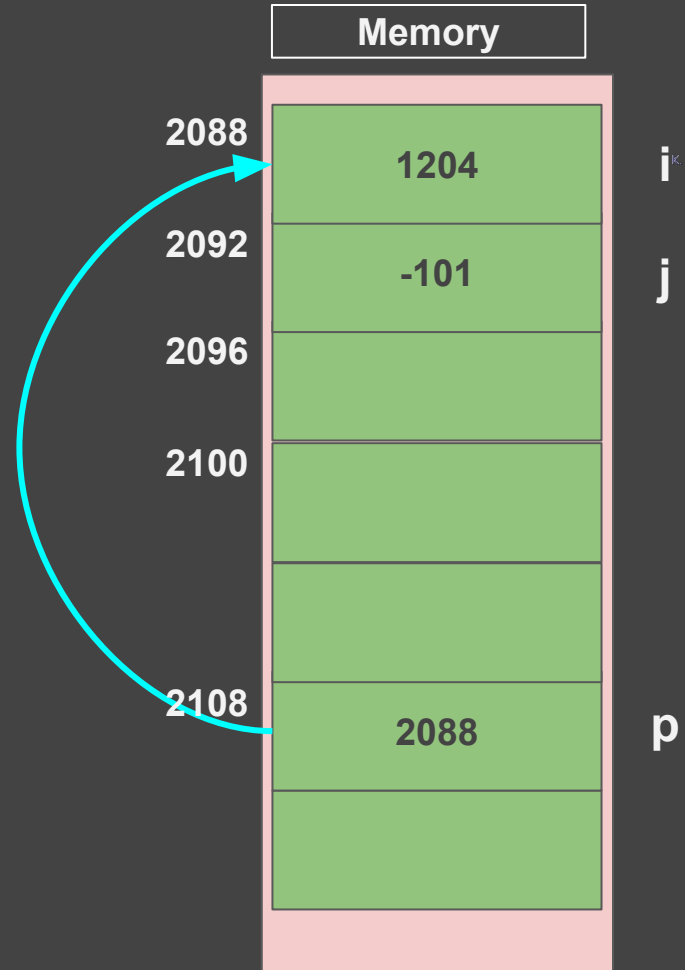
1926

K. J. Somaiya Institute of Technology

What is a Pointer?

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

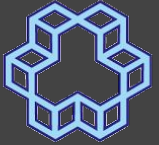
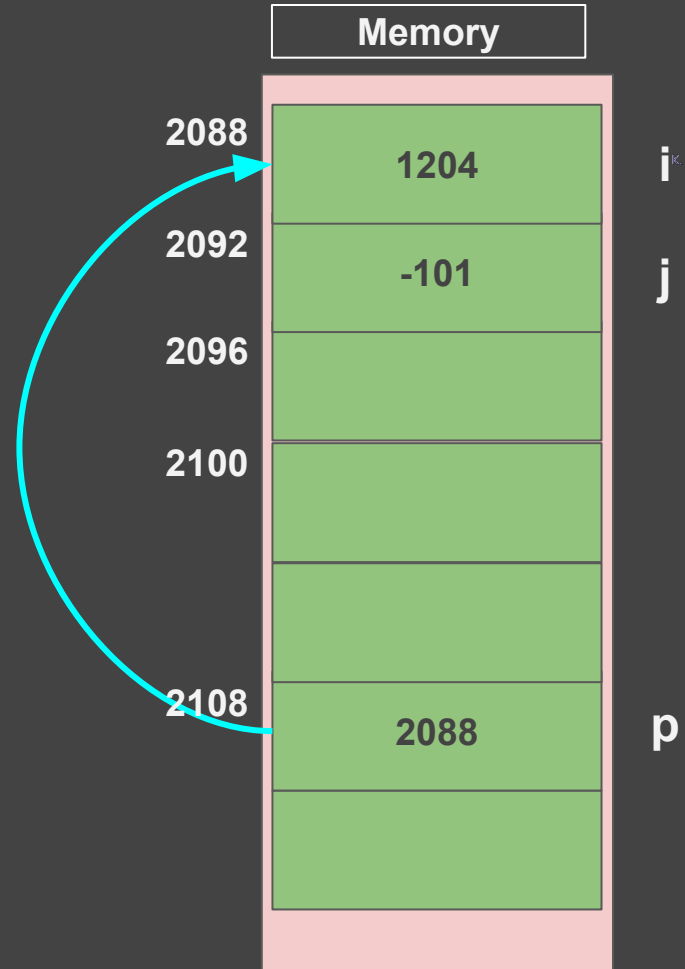
i	p	&i	&p
1204	2088	2088	?



What is a Pointer?

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

i	p	&i	&p
1204	2088	2088	2108



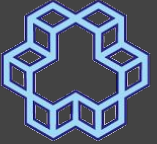
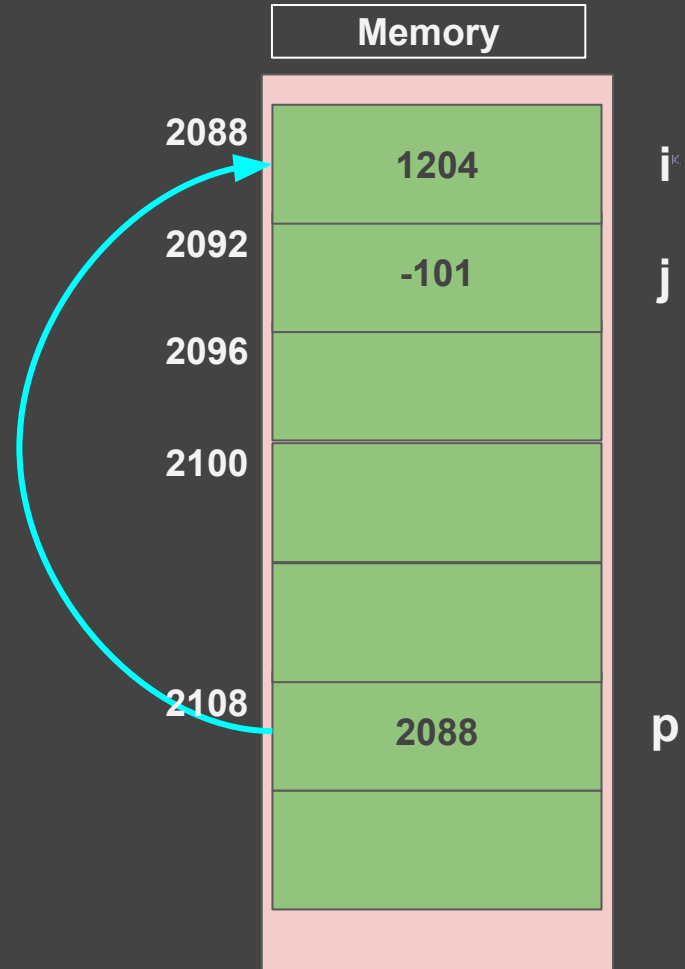
1926

K. J. Somaiya Institute of Technology

What is a Pointer?

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

i	p	&i	&p
1204	2088	2088	2108



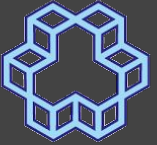
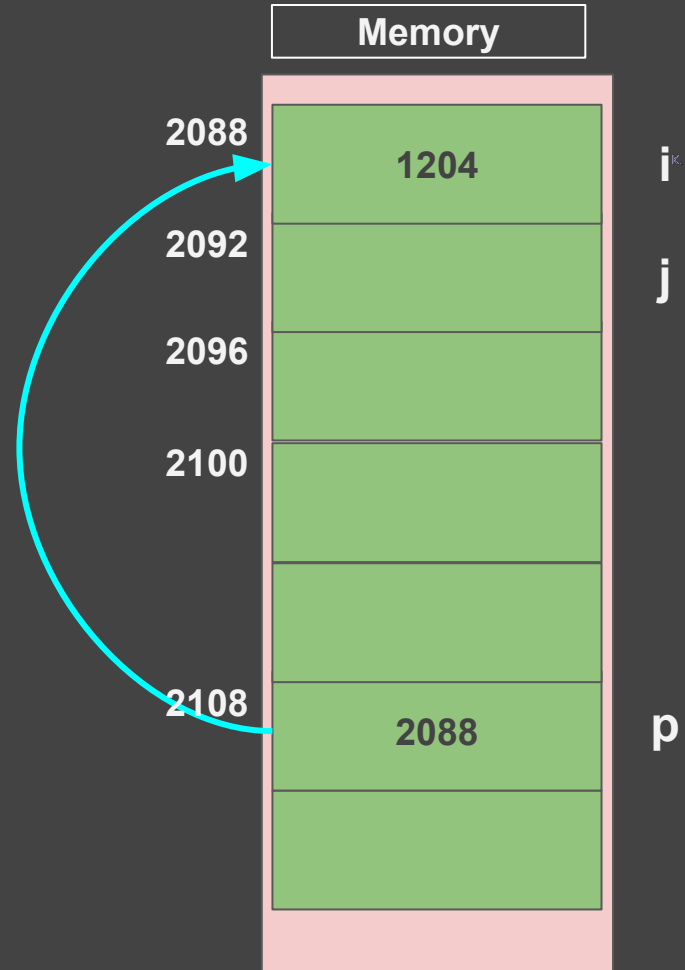
1926

K. J. Somaiya Institute of Technology

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	?



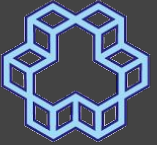
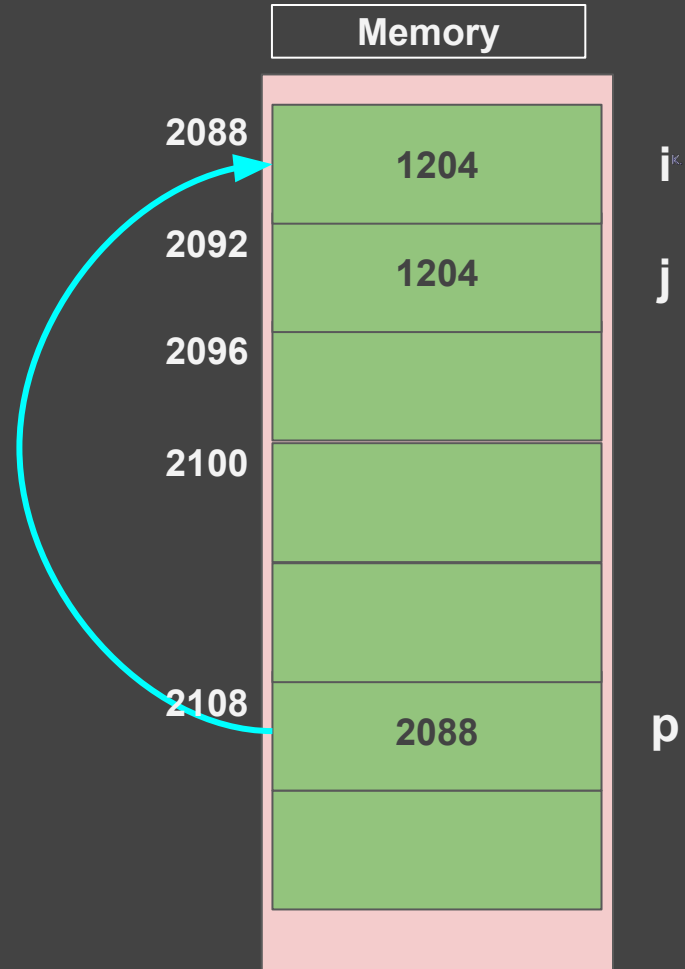
1926

K. J. Somaiya Institute of Technology

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



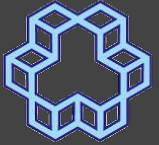
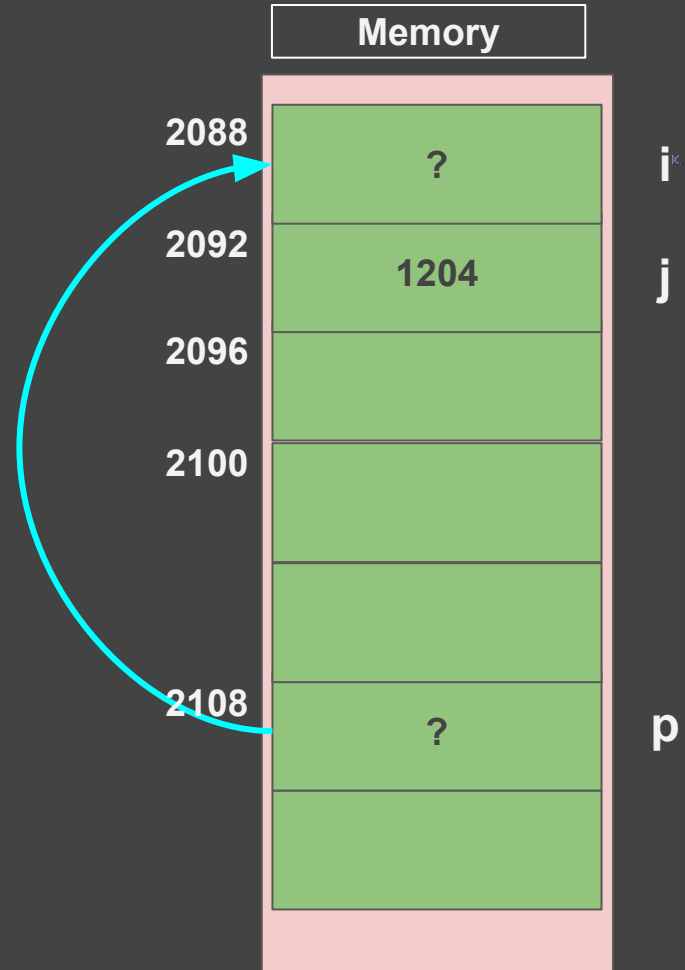
1926

K. J. Somaiya Institute of Technology

What is a Pointer?

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

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



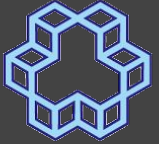
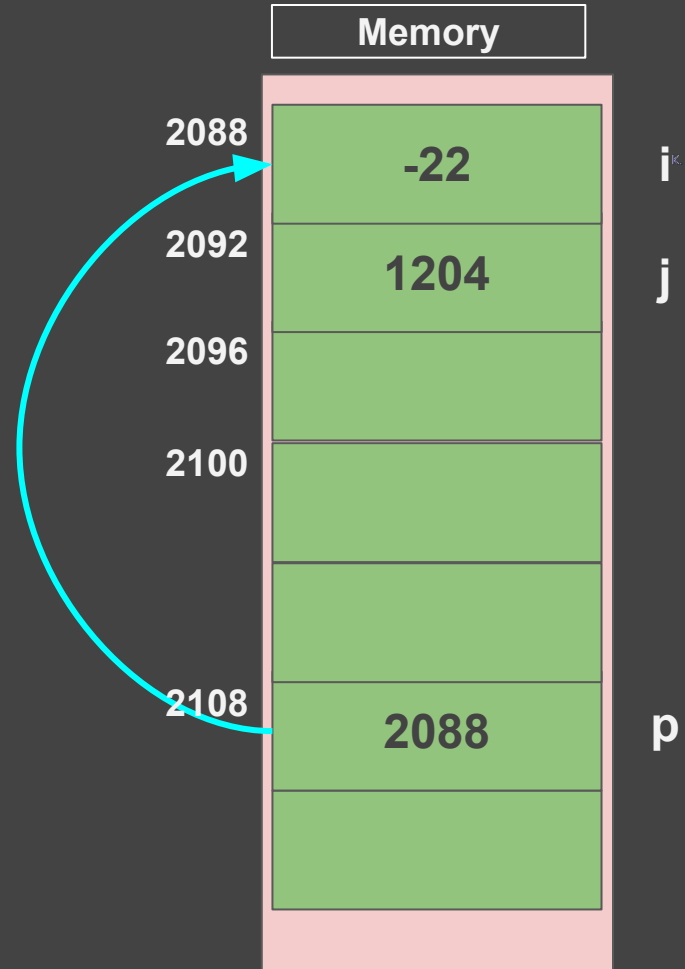
1926

K. J. Somaiya Institute of Technology

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

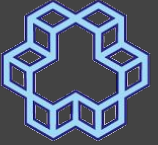


1926

K. J. Somaiya Institute of Technology

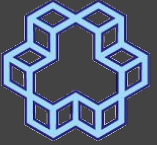
Argument passing

- By value
- By reference



1926

K. J. Somaiya Institute of Technology



1926

K. N. Toosi University of Technology

Simulating by-reference argument passing

```
#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 *= 2;
}
```



1926

K. N. Toosi University of Technology

Simulating by-reference argument passing

```
#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 *= 2;
}
```

```
#include <stdio.h>

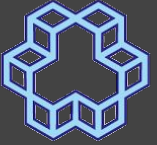
void duplicate(int*);

int main() {
    int x = 10;

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

    return 0;
}

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



1926

K. N. Toosi University of Technology

Simulating by-reference arguments

```
#include <stdio.h>

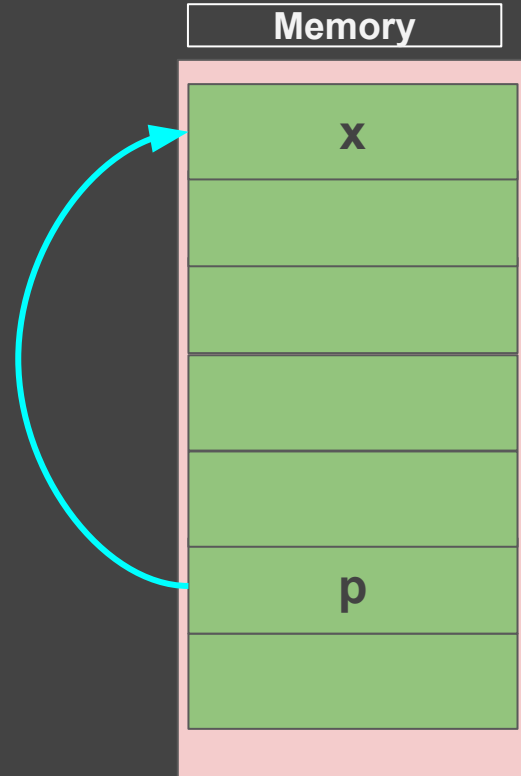
void duplicate(int*);

int main() {
    int x = 10;

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

    return 0;
}

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



multiple outputs

```
#include <stdio.h>

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

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

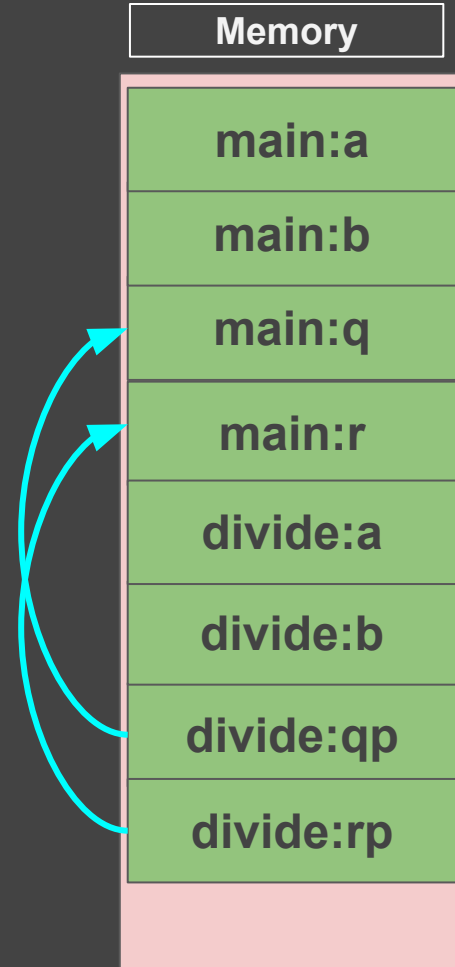
    scanf("%d %d", &a, &b);

    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;
}
```



1926

K. N. Toosi University of Technology

Arguments are still by-value

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

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

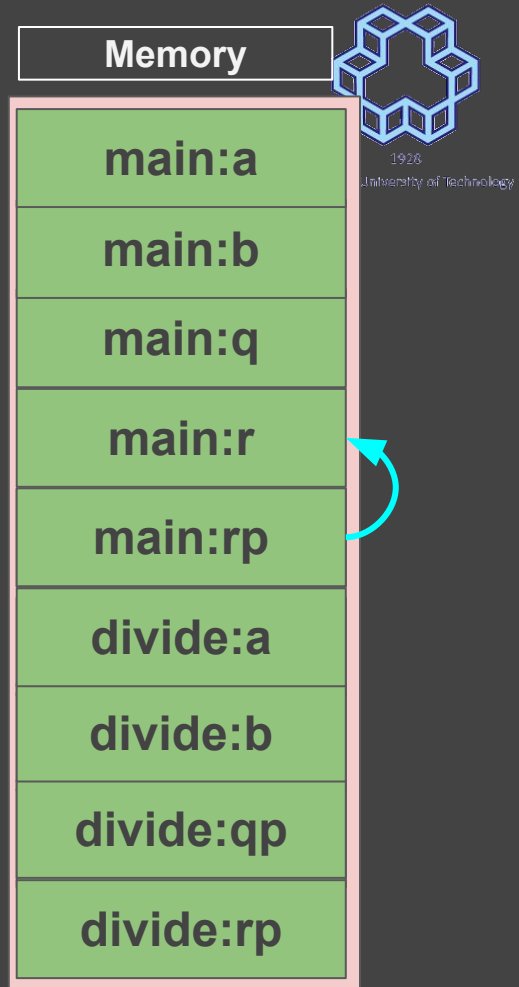
    rp = &r;
    divide(a,b,&q,rp);

    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

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

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

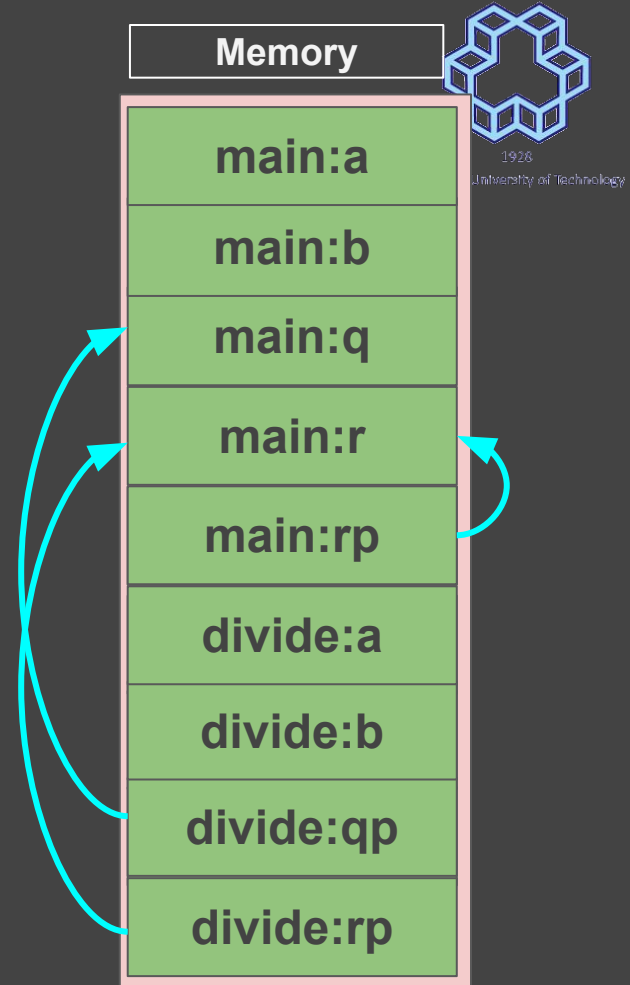
    rp = &r;
    divide(a,b,&q,rp);

    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

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

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

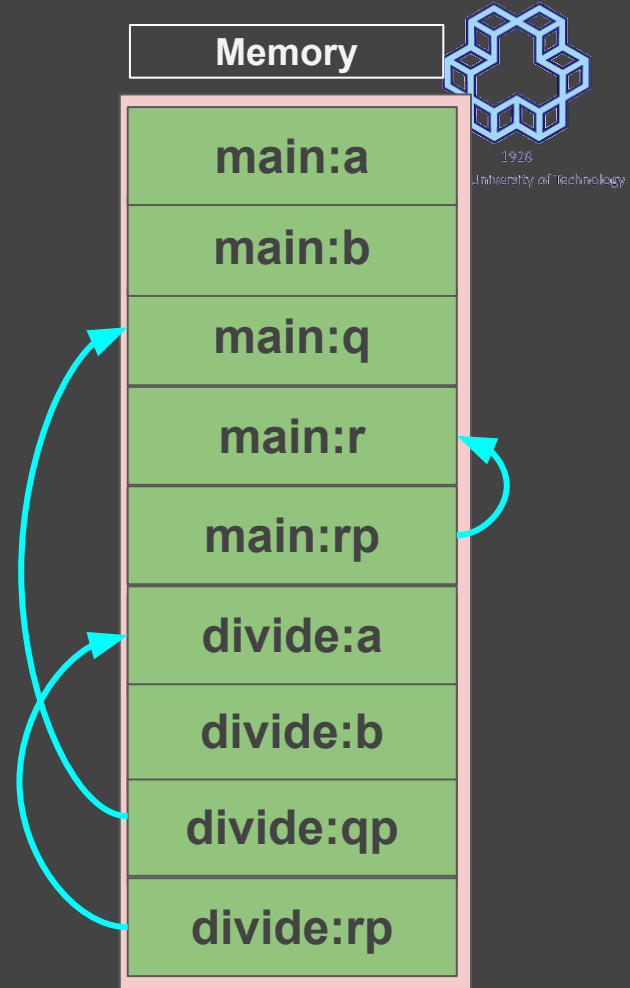
    rp = &r;
    divide(a,b,&q,rp);

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

    return 0;
}

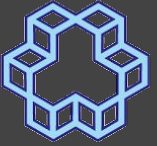
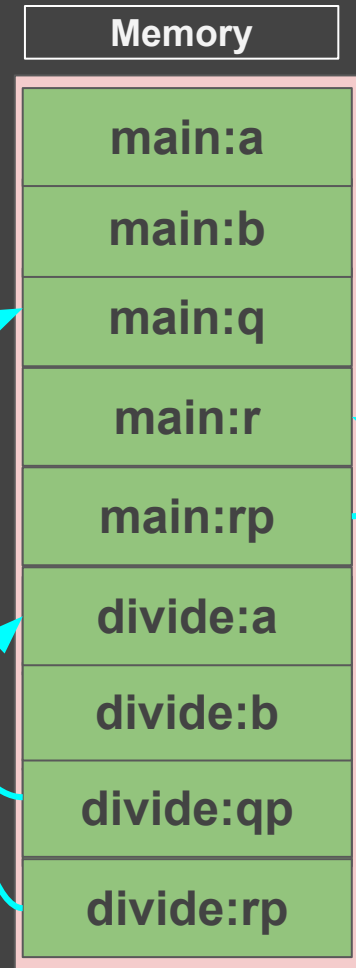
void divide(int a, int b, int *qp, int *rp) {
    *qp = a / b;
    *rp = a % b;
    rp = &a;

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



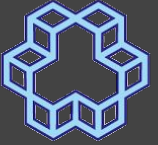
use %p to print addresses

```
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);  
}
```



1926

K. N. Toosi University of Technology



1926

K. N. Toosi University of Technology

Example: Swapping

```
#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);

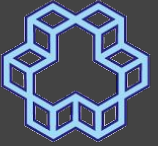
    printf("x=%.1f, y=%.1f\n", x, y);

    return 0;
}
```

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

pointerfunc5_swap.c

Pointers



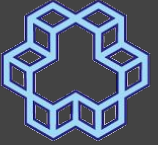
1926

K. J. Somaiya Institute of Technology



```
*y = 13;
```

constant pointers



1926

K. N. Toosi University of Technology

```
#include <stdio.h>

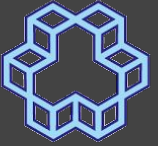
int main() {
    int a=0, b=1;

    int *p = &a;

    p = &b;

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

constant pointers



1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

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

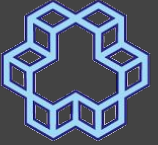
    int *p = &a;

    p = &b;

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

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

constant pointers



1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

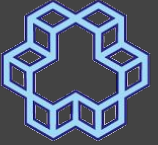
int main() {
    int a=0, b=1;

    const int *p = &a;

    p = &b;

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

constant pointers



1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

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

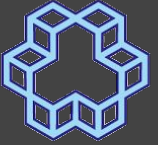
    const int *p = &a;

    p = &b;

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

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

constant pointers



1926

K. J. Somaiya Institute of Technology

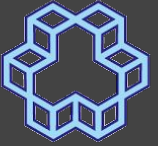
```
#include <stdio.h>

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

    const int *p = &a;

    *p = 2;

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

1926

K. J. Somaiya Institute of Technology

constant pointers

```
#include <stdio.h>

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

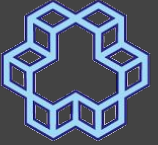
    const int *p = &a;

    *p = 2;

    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



1926

K. N. Toosi University of Technology

```
#include <stdio.h>

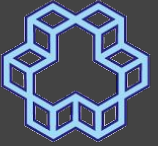
int main() {
    int a=0, b=1;

    int * const p = &a;

    *p = 2;

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

constant pointers



1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

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

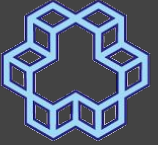
    int * const p = &a;

    *p = 2;

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

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

constant pointers



1926

K. N. Toosi University of Technology

```
#include <stdio.h>

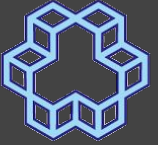
void swap(double *, double *);

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

    int * const p = &a;

    p = &b;

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



1926

K. N. Toosi University of Technology

constant pointers

```
#include <stdio.h>

void swap(double *, double *);

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

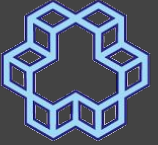
    int * const p = &a;

    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



1926

K. N. Toosi University of Technology

```
#include <stdio.h>

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

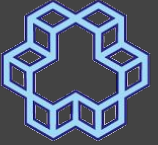
    const int * const p = &a;

    *p = 2;

    p = &b;

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

constant pointers



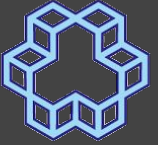
1926

K. J. Somaiya Institute of Technology

same thing

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

Pointers



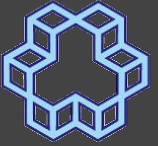
1926

K. J. Somaiya Institute of Technology

```
a = *(&b) ;
```

```
p = &(*q) ;
```

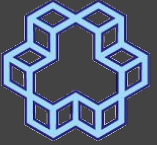

Remember: the "swap" function



1926

K. N. Toosi University of Technology

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



1926

K. J. Somaiya Institute of Technology

Remember: bubble sort

```
int main() {  
    int a[] = {25,19,14,18,27,6,32,18,20,1,21};  
    int n = sizeof(a) / sizeof(a[0]);  
  
    printArray(a,n);  
  
    bubbleSort(a,n);  
  
    printArray(a,n);  
  
    return 0;  
}
```

```
nasihatkon@kntu:code$ gcc pointersort.c && ./a.out  
25, 19, 14, 18, 27, 6, 32, 18, 20, 1, 21,  
1, 6, 14, 18, 18, 19, 20, 21, 25, 27, 32,  
nasihatkon@kntu:code$
```

```
void bubbleSort(int a[], int n) {  
    int m;  
  
    for (m = n-1; m > 0; m--) {  
        for (int i = 0; i < m; i++) {  
  
            if (a[i] > a[i+1]) {  
                int temp = a[i];  
                a[i] = a[i+1];  
                a[i+1] = temp;  
            }  
  
        }  
    }  
}
```



1926

K. J. Somaiya Institute of Technology

Using swap for bubble sort

```
void bubbleSort(int a[], int n) {
    int m;

    for (m = n-1; m > 0; m--) {
        for (int i = 0; i < m; i++) {

            if (a[i] > a[i+1]) {
                int temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }

        }
    }
}
```

```
void bubbleSort(int a[], int n) {
    int m;

    for (m = n-1; m > 0; m--) {
        for (int i = 0; i < m; i++) {

            if (a[i] > a[i+1]) {
                swap(&a[i], &a[i+1]);
            }

        }
    }
}
```

pointersort.c

```
nasihatkon@kntu:code$ gcc pointersort.c && ./a.out
25, 19, 14, 18, 27, 6, 32, 18, 20, 1, 21,
1, 6, 14, 18, 18, 19, 20, 21, 25, 27, 32,
nasihatkon@kntu:code$
```