Pointers

Nalini Vasudevan
Columbia University

Variables

- Allocate 1 byte of memory

```
char a;
```

- Memory allocated at some particular address

```
char *ptr;
ptr = &a;
```

- All pointers are of the same size
    - they hold the address
    - generally 4 bytes

More Pointers

```c
int x = 1, y = 2, z[10];
int *ip;    /* ip is a pointer to int */

ip = &x;    /* ip now points to x */
y = *ip;    /* y is now 1 */
*ip = 0;    /* x is now 0 */
ip = &z[0]; /* ip now points to z[0] */
```

Quiz

```
int x = 1, y = 2, z[10];
int *ip;
ip = &x;
*ip = *ip + 1;
printf("%d %d %d", x, y, *ip);
y = *ip + 1;
printf("%d %d %d", x, y, *ip);
*ip += 1;
printf("%d %d %d", x, y, *ip);
```

Quiz

```c
int x = 1, y = 2, z[10];
int *ip;
ip = &x;
*ip = *ip + 1;
printf("%d %d %d", x, y, *ip); /* 2 2 2 */
y = *ip + 1;
printf("%d %d %d", x, y, *ip); /* 2 3 2 */
*ip += 1;
printf("%d %d %d", x, y, *ip); /* 3 3 3 */
```

Swap two numbers

```c
void swap (int x, int y) /*wrong*/
{
        int temp;

        temp = x;
        x = y;
        y = temp;

}
main()
{
        int a = 5, b = 3;
        swap (a, b);
        printf ("After swap: %d %d", a, b);

}
```

Swap two numbers

```c
void swap (int *px, int *py)
{
        int temp;

        temp = *px;
        *px = *py;
        *py = temp;

}
main()
{

        int a = 5, b = 3;
        swap (&a, &b);
        printf ("After swap: %d %d", a, b);

}
```

Arrays and Pointers

```
main()
{
        int a[4] = {11, 12, 13, 14};
        int x, y;
        int *pa;
        pa = &a[0];
        x = *pa;
        y = *(pa + 2);
        y = y + 3;
        printf ("%d %d %d %d", *pa, x,
y, a[2]);
}
```

Arrays and Pointers

```c
main()
{
        int a[4] = {11, 12, 13, 14};
        int x, y;
        int *pa;
        pa = &a[0];
        x = *pa;
        y = *(pa + 2);
        y = y + 3;
        printf ("%d %d %d %d", *pa,
x, y, a[2]); /* 11, 11, 16, 13 */
}
```

Arrays and Pointers

pa = a;

is equivalent to

pa = &a[0];

x = a[i];

is equivalent to

x= *(a + i);

Arrays and Pointers

```
int mystrlen (char s[])
{
    int i, len = 0;
    for (i = 0; s[i] != '\0'; i++)
            len++;
    return len;

}
main()
{
    char str[10];
    strcpy(str, "Nalini");
    printf("%d", mystrlen(str));

}
```

Arrays and Pointers

```c
int mystrlen (char *s)
{
    int len = 0;
    for (; *s != '\0'; s++)
            len++;
    return len;
}
main()
{
    char str[10];
    strcpy(str, "Nalini");
    printf("%d", mystrlen(str));
}
```

Arrays and Pointers

```c
int mystrlen (char *s)
{
    int len = 0;
    for (; *s != '\0'; s++)
            len++;
    return len;
}
main()
{
    char str[10];
    scanf("%s", str);
    printf("%d", mystrlen(str));
}
```

Array of Pointers

```c
int main()
{
        int x = 11, y = 12, z = 13, i;
        int *a[3];
        a[0] = &x;
        a[1] = &y;
        a[2] = &z;
        for  (i = 0; i < 3; i++)
                printf("%d\n", *a[i]);
}
```

Pointers to functions

```c
int sum(int a, int b)
{
        int res = a + b;
        return res;
}
main()
{
    int z;
    int (*fp)(int, int);
    fp  = sum;
    z = (*fp)(3, 4);
    z = sum (3, 4);
}
```