

Arrays

- **Array: Typed collection of values indexed by nonnegative integers.**
- **Type and size must be known at compile time, except in the case of formal parameters:**
`int a[20]; void sort(int b[], int n);`
- **Arrays in C are stored one-dimensionally:**
`piece chess[8][8];` (chess has type piece *)
- **Strings are arrays of characters:**
`char msg[21] = "Enter your password:";`

20

Pointers and Arrays

- **An array name is considered pointer to first element:** `int a[5];`
 - `a` is pointer to `a[0]`
 - `pa = &a[0]` and `pa = a` mean the same thing
 - `a+1` means L-value of `a[0]` plus as many bytes as are needed to store value of elements of `a`'s type
 - Pointer arithmetic is an address calculation with respect to the underlying representation
- **An array name is a constant pointer**
 - `a++` and `a = pa` are illegal

21

Arrays and Pointers

- **Pointer Arithmetic:**

```
int j = 5; int *k = &j;
```

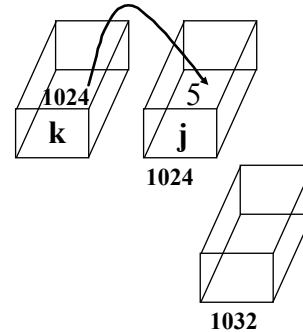
$(*k+2)$ means $((*k)+2)$:

legal R-value, illegal L-value

$(k+2)$: legal R-value, illegal L-value

$*(k+2)$: legal R-value, legal L-value

(perhaps not meaningful)



- **General Rule:**

Given declaration $T \ *ptr$;

$(ptr + k)$ points to location $ptr + k \times \text{sizeof}(T)$

22

Arrays and Pointers

- **Two ways to process the array $A[20]$**

– with subscripts:

```
int k;  
for(k=0; k < 20; k++){  
    A[k] = 0;  
}
```

– with pointers:

```
int *ap;  
for(ap=A; ap < A+20; ap++){  
    *ap = 0;  
}
```

23

Strings and Pointers

- The library package `string.h` assumes that a string is an array of characters terminated by `'\0'`.

- Thus:

```
char name[5] = "Mary";
```

- Example:

```
int strlen(char const *str)
{
    int n=0;
    for( ; *str != '\0' ; str++) { n+=1;}
    return n;
}
```

24

Strings and Pointers

- Example:

```
char *strcpy(char *dst, char const *src)
{
    char *str = dst;
    for( ; *src != '\0' ; src++, dst++)
        {*dst = *src;}
    *dst = '\0';
    return str;
}
```

- **Caution:** Programmer must guarantee that `dst` has been allocated enough memory to store `src`

25