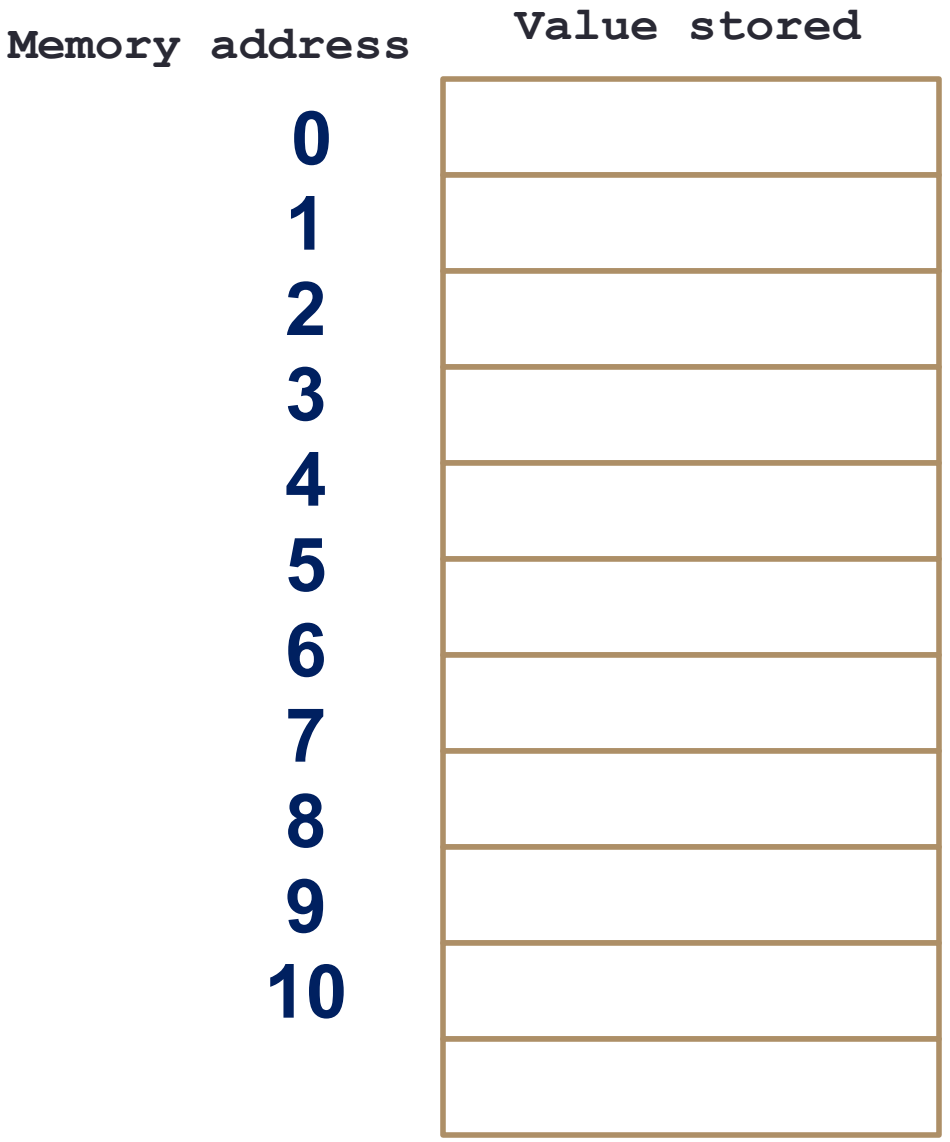
C++ ARRAYS
NUMBER CONVERSIONS

Problem Solving with Computers-I

General model of memory

- Sequence of adjacent cells
- Each cell has 1-byte stored in it
- Each cell has an address (memory location)



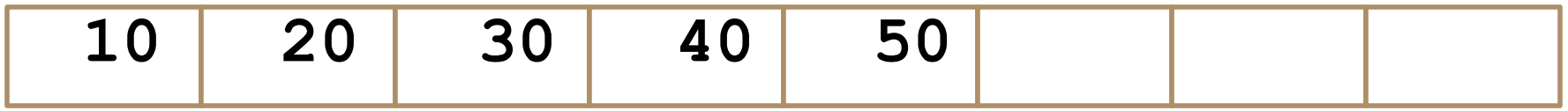
Storing sequences in programs

Write a program to take a sequence of midterm scores (out of 100) and compute the average of the midterm

C++ Arrays

A C++ array is a **list of elements** that share the same name, have the same data type and are located adjacent to each other in memory

scores



Declare:

Exercise: Reassign each value to 60


```
int scores[]={20,10,50}; // declare an initialize
//Access each element and reassign its value to 60
```

Exercise: Increment each element by 10


```
int scores[]={20,10,50}; // declare an initialize
//Increment each element by 10
```

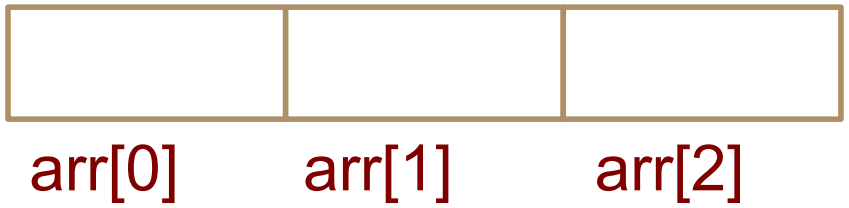
Most common array pitfall- out of bound access

scores[0] scores[1] scores[2]

```
int arr[]={20,10,50}; // declare an initialize
for(int i=0; i<=3; i++)
    scores[i] = scores[i]+10;
```

Demo: Passing arrays to functions

Tracing code involving arrays

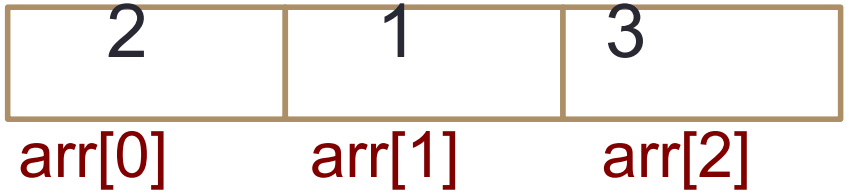



```
int arr[]={1,2,3};
int tmp = arr[0];
arr[0] = arr[2];
arr[2] = tmp;
```

Choose the resulting array after the code is executed

A.

B.



C.

D. None of the above

What is the memory location of each element?


```
int scores[]={10, 20, 30, 40, 50};
```

If the starting location of the array is 0x200, what is memory location of element at index 2?

A. 0x201

B. 0x202

C. 0x204

D. 0x208

Converting between binary and decimal

Binary to decimal: $1\ 0\ 1\ 1\ 0_2 = ?_{10}$

Decimal to binary: $34_{10} = ?_2$

Hex to binary

- Each hex digit corresponds directly to four binary digits
- Programmers love hex, why?
- Convert to binary

0x25B= ?

00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Hexadecimal to decimal

$25B_{16}$ = ? Decimal

Hexadecimal to decimal

- Use polynomial expansion
- $25B_{16}$ = 2*256 + 5*16 + 11*1 = 512 + 80 + 11
  = 603
- Decimal to hex: $36_{10}=?_{16}$

Binary to hex: 1000111100

A. 8F0

B. 23C

C. None of the above

BIG IDEA: Bits can represent anything!!

Numbers	Binary Code	Colors	Binary code
0		Red	
1			
2		Green	
3		Blue	

N bits can represent at most 2^N things

What is the minimum number of bits required to represent all the letters in the English alphabet (assume only upper case)?

A. 3
B. 4
C. 5
D. 6
E. 26

What is the maximum positive value that can be stored in a byte?

A. 127

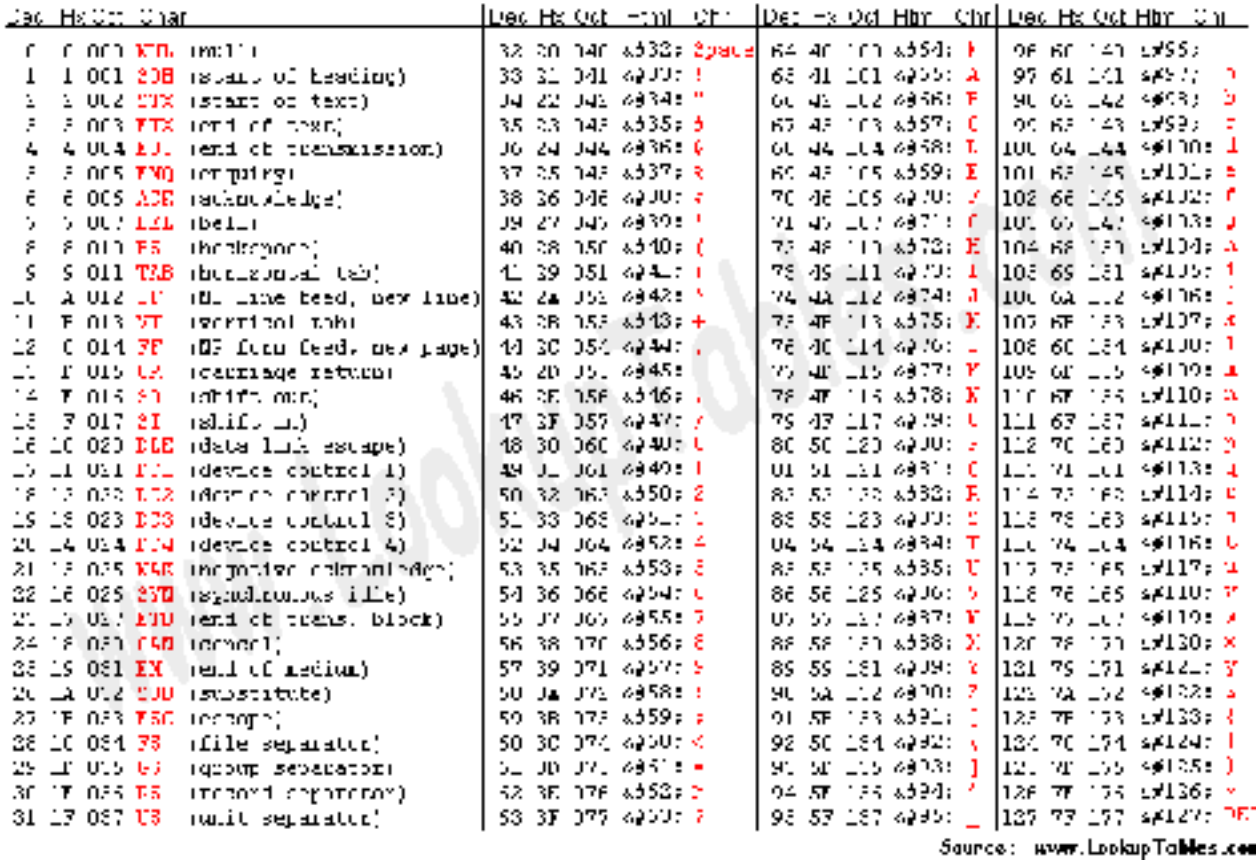
B. 128

C. 255

D. 256

BIG IDEA: Bits can represent anything!!

- Logical values?
  - 0 ⇒ False, 1 ⇒ True
- colors ?
- Characters?
  - 26 letters ⇒ 5 bits ($2^5 = 32$)
  - upper/lower case + punctuation ⇒ 7 bits (in 8) (“ASCII”)
  - standard code to cover all the world’s languages ⇒ 8,16,32 bits (“Unicode”) www.unicode.com
- locations / addresses? commands?



Source: www.LookupTables.com

ASCII table

- REMEMBER: N bits ⇔ at most 2^N things

Next time

- Pointers
- Mechanics of function calls – call by value and call by reference