

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)

Memory address	Value stored
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

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

10	20	30	40	50			
----	----	----	----	----	--	--	--

Declare:

What is the memory location of each element?

scores

10	20	30	40	50
----	----	----	----	----

```
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

Exercise: Reassign each value to 60



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

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

Exercise: Increment each element by 10



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

```
int scores[]={20,10,50}; // declare and 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;
```

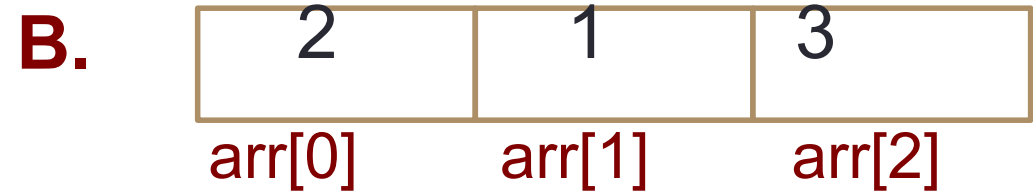
Demo: Passing arrays to functions

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



D. None of the above

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

$$25B_{16} = ? \text{ Decimal}$$

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		 <i>Red</i>	
1			
2		 <i>Green</i>	
3		 <i>Blue</i>	

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

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	000	NUL	128	80	140	end of text	256	100	160	end of text	384	180	240	end of text
1	1	001	SOH	129	81	141	start of heading	257	101	161	start of heading	385	181	241	start of heading
2	2	010	STX	130	82	142	start of text	258	102	162	start of text	386	182	242	start of text
3	3	011	ETX	131	83	143	end of text	259	103	163	end of text	387	183	243	end of text
4	4	100	TX	132	84	144	end of transmission	260	104	164	end of transmission	388	184	244	end of transmission
5	5	101	TX	133	85	145	comparing	261	105	165	comparing	389	185	245	comparing
6	6	110	ACK	134	86	146	acknowledge	262	106	166	acknowledge	390	186	246	acknowledge
7	7	111	NAK	135	87	147	negative acknowledge	263	107	167	negative acknowledge	391	187	247	negative acknowledge
8	8	010	FS	136	88	148	file separator	264	108	168	file separator	392	188	248	file separator
9	9	011	TSB	137	89	149	tab	265	109	169	tab	393	189	249	tab
10	A	100	LF	138	8A	150	line feed, new line	266	110	170	line feed, new line	394	190	250	line feed, new line
11	B	101	VT	139	8B	151	vertical tab	267	111	171	vertical tab	395	191	251	vertical tab
12	C	010	FF	140	8C	152	form feed, new page	268	112	172	form feed, new page	396	192	252	form feed, new page
13	D	011	CR	141	8D	153	carriage return	269	113	173	carriage return	397	193	253	carriage return
14	E	100	SH	142	8E	154	shift out	270	114	174	shift out	398	194	254	shift out
15	F	101	SI	143	8F	155	shift in	271	115	175	shift in	399	195	255	shift in
16	10	000	DEL	144	90	156	delete (backspace)	272	116	176	delete (backspace)	400	196	256	delete (backspace)
17	11	001	DC1	145	91	157	device control 1	273	117	177	device control 1	401	197	257	device control 1
18	12	010	DC2	146	92	158	device control 2	274	118	178	device control 2	402	198	258	device control 2
19	13	011	DC3	147	93	159	device control 3	275	119	179	device control 3	403	199	259	device control 3
20	14	100	DC4	148	94	160	device control 4	276	120	180	device control 4	404	200	260	device control 4
21	15	101	NAK	149	95	161	negative acknowledge	277	121	181	negative acknowledge	405	201	261	negative acknowledge
22	16	000	SYN	150	96	162	synchronous idle	278	122	182	synchronous idle	406	202	262	synchronous idle
23	17	001	TX	151	97	163	end of transmission block	279	123	183	end of transmission block	407	203	263	end of transmission block
24	18	010	TX	152	98	164	cancel	280	124	184	cancel	408	204	264	cancel
25	19	011	TX	153	99	165	end of medium	281	125	185	end of medium	409	205	265	end of medium
26	1A	100	TX	154	9A	166	substitute	282	126	186	substitute	410	206	266	substitute
27	1B	101	TX	155	9B	167	escape	283	127	187	escape	411	207	267	escape
28	1C	000	TX	156	9C	168	file separator	284	128	188	file separator	412	208	268	file separator
29	1D	001	TX	157	9D	169	group separator	285	129	189	group separator	413	209	269	group separator
30	1E	010	TX	158	9E	170	private separator	286	130	190	private separator	414	210	270	private separator
31	1F	011	TX	159	9F	171	unit separator	287	131	191	unit separator	415	211	271	unit separator

Source: www.LookupTables.com

ASCII table

- **REMEMBER:** N bits \Leftrightarrow at most 2^N things

Next time

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