CSC180: Lecture 20

Wael Aboulsaadat

wael@cs.toronto.edu

http://portal.utoronto.ca/

Acknowledgement: These slides are partially based on the slides supplied with Prof. Savitch book: Problem Solving with C

Revision

Retuning largest value in an array

- Pseudo-code?
    - Assume first element is the largest
    - Visit all elements starting from second and do the following
        - if it is larger than the largest
            - Set largest to it.

Retuning largest value in an array

```
int returnLargestNumber( const int narrInput[] , int nSize )
{
    int nIndex,nLargest;

    if( nSize == 0 )
        return -9999999;

    nLargest = narrInput[0];
    for( nIndex = 1; nIndex < nSize; nIndex  = nIndex + 1 )
        if( narrInput[nIndex] > nLargest )
                nLargest = narrInput[nIndex];

    return nLargest;
}
```

Finding largest value in an array (using recursion)

- Pseudo-code?

  base condition: we are done with the array no more elements to visit ? When is that? Some index variable is == to array size

  recursive case: compare current element against largest and call function recursively

Finding largest value in an array (using recursion)

```
int  returnLargestNumber( const int narrInput[] , int nSize ,
                                         int& nIndex )
 {

          // Can you write this?

 }
```

Given an array, replace all occurrences of a value with another

- Pseudo-code?
    - Visit all elements starting from first and do the following
        - if it is the value we are looking for then
            - Change it to the new value

Given an array, replace all occurrences of a value with another

```
void replaceNumber( int narrInput[] , int nSize,
                    int nOld, int nNew )
{
    int nIndex;
    for( nIndex = 0; nIndex < nSize; nIndex = nIndex + 1 )
        if( narrInput[nIndex] == nOld )
            narrInput[nIndex] = nNew;
}
```

Given an array, replace all occurrences of a value with another (using pointers)

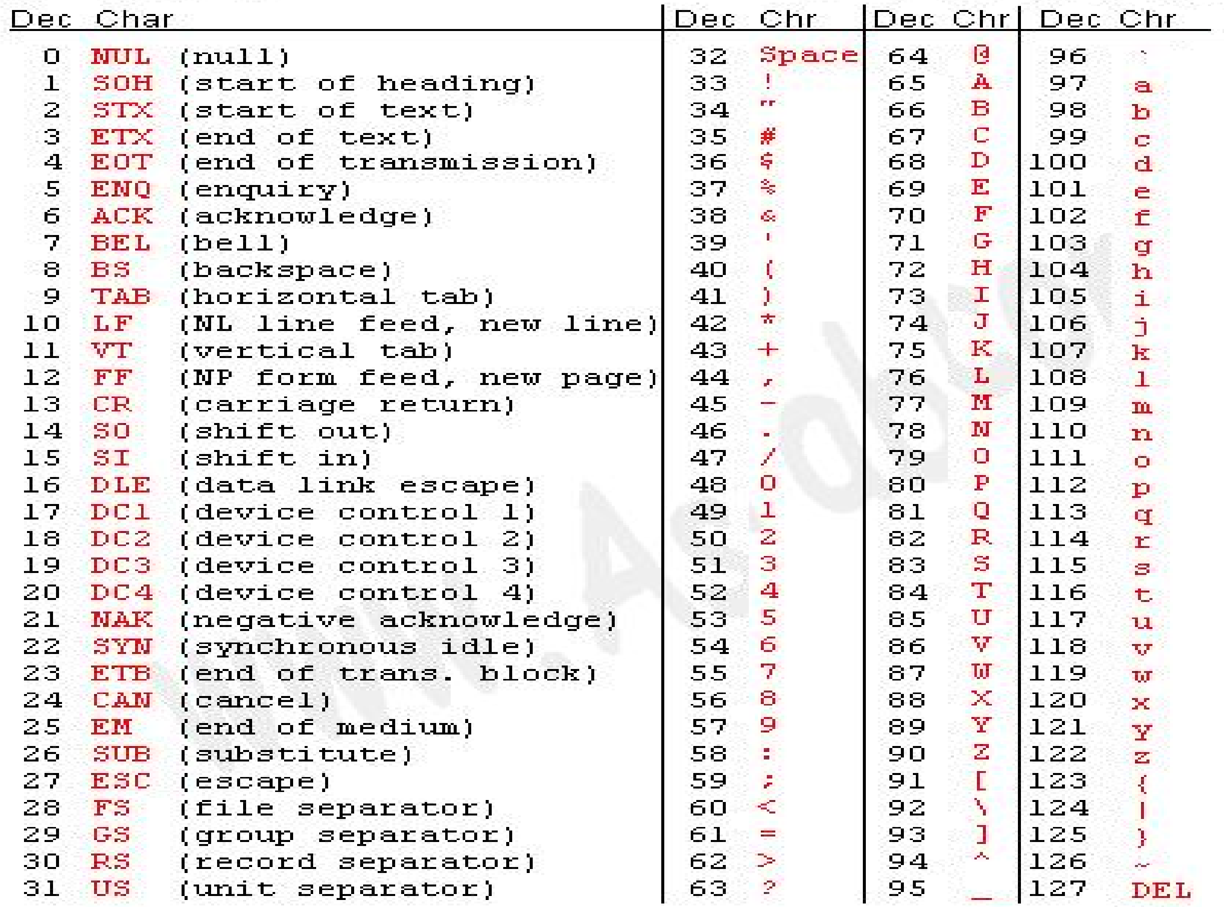
```
void replaceNumber( int narrInput[] , int nSize,
                                   int nOld, int nNew )
{
    int *nptrVal;

    for( nptrVal = &narrInput[0]; nptrVal <= &narrInput[nSize-1] ;
                                                  nptrVal = nptrVal + 1)
          if( *nptrVal == nOld )
                    *nptrVal = nNew;
}
```

Given an array of characters, change all lowercase to uppercase

- Given: Ascii of a is 97 while Ascii of A is 65,….

  Ascii of z is 122 while Ascii of Z is 90



Dec	Char		Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	(null)	32	Space	64	@	96	`
1	SOH	(start of heading)	33	!	65	A	97	a
2	STX	(start of text)	34	"	66	B	98	b
3	ETX	(end of text)	35	#	67	C	99	c
4	EOT	(end of transmission)	36	$	68	D	100	d
5	ENQ	(enquiry)	37	%	69	E	101	e
6	ACK	(acknowledge)	38	&	70	F	102	f
7	BEL	(bell)	39	'	71	G	103	g
8	BS	(backspace)	40	(	72	H	104	h
9	TAB	(horizontal tab)	41	)	73	I	105	i
10	LF	(NL line feed, new line)	42	*	74	J	106	j
11	VT	(vertical tab)	43	+	75	K	107	k
12	FF	(NP form feed, new page)	44	,	76	L	108	l
13	CR	(carriage return)	45	-	77	M	109	m
14	SO	(shift out)	46	.	78	N	110	n
15	SI	(shift in)	47	/	79	O	111	o
16	DLE	(data link escape)	48	0	80	P	112	p
17	DC1	(device control 1)	49	1	81	Q	113	q
18	DC2	(device control 2)	50	2	82	R	114	r
19	DC3	(device control 3)	51	3	83	S	115	s
20	DC4	(device control 4)	52	4	84	T	116	t
21	NAK	(negative acknowledge)	53	5	85	U	117	u
22	SYN	(synchronous idle)	54	6	86	V	118	v
23	ETB	(end of trans. block)	55	7	87	W	119	w
24	CAN	(cancel)	56	8	88	X	120	x
25	EM	(end of medium)	57	9	89	Y	121	y
26	SUB	(substitute)	58	:	90	Z	122	z
27	ESC	(escape)	59	;	91	[	123	{
28	FS	(file separator)	60	<	92	\	124	|
29	GS	(group separator)	61	=	93	]	125	}
30	RS	(record separator)	62	>	94	^	126	~
31	US	(unit separator)	63	?	95	_	127	DEL

Given an array of characters, change all lowercase to uppercase

- Pseudo-code?

```
Visit every character
if it is a small letter (i.e. ascii between 97 and 22)
    subtract 32 from the ascii value
```

Given an array of characters, change all lowercase to uppercase

```
void changetoUpperCase( char carrInput[] )
{
    int nIndex;
    int nChar;

    nIndex = 0;
    while( carrInput[nIndex] != '\0' )
    {
        nChar = carrInput[nIndex];
        if( nChar >= 97 && nChar <= 122 )
                carrInput[nIndex] = carrInput[nIndex]  - 32;
        nIndex = nIndex + 1;
    }
}
```

Note:
This function assumes that the passed char array ends with ' \0 ‘, so we loop till we hit it

Additional Exercises

- *Write a function that calculates Fibonacci value recursively*

$$F_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F_{n-1} + F_{n-2} & \text{if } n > 1. \end{cases}$$

- *Write a function that multiply two matrices. Each matrix is represented as a 2D array.*

- *Rewrite the binary search using pointers instead of indices*