Pointers to Arrays

```
/* Declarations */
char greeting[] = "Hello World.";
char question[] = "How are you?";
char buffer[80];
char *pCharArray = greeting; /* Pointer points to greeting[0] */

/* Making the pointer point to different locations */
pCharArray = &greeting[2];   /* Pointer points to greeting[2] */
pCharArray++;                /* Pointer points to greeting[3] */
pCharArray--;                /* Pointer points to greeting[2] */
--pCharArray;                /* Pointer points to greeting[1] */
++pCharArray;                /* Pointer points to greeting[2] */
pCharArray = &question[5];   /* Pointer points to question[5] */
++pCharArray;                /* Pointer points to question[6] */
pCharArray++;                /* Pointer points to question[7] */
--pCharArray;                /* Pointer points to question[6] */
pCharArray--;                /* Pointer points to question[5] */
pCharArray = question;       /* Pointer points to question[0] */
pCharArray += 5;             /* Pointer points to question[5] */
pCharArray -= 3;             /* Pointer points to question[2] */

/* Printing the character (dereferencing the pointer) */
printf("%c", *pCharArray);       /* prints question[2], (at 2) */
printf("%c", *(pCharArray + 2)); /* prints question[4], (2+2)  */
printf("%c", pCharArray[4]);     /* prints question[6], (2+4)  */
```

Sample Run,

```
z:\apc-cc-conduit\images>a
wae
z:\apc-cc-conduit\images>
```

```
/* Printing out the string (from current position to NULL) */
printf("%s", pCharArray);
```

Sample Run,

```
z:\apc-cc-conduit\images>a
w are you?
z:\apc-cc-conduit\images>
```

```
/* Using the pointer with string functions */
strcpy(buffer, pCharArray);
printf("1: %s\n2: %s", buffer, pCharArray);
```

Sample Run,

```
z:\apc-cc-conduit\images>a
1: w are you?
2: w are you?
z:\apc-cc-conduit\images>
```

```
/* Changing the character that the pointer points to */
pCharArray = greeting;    /* Points to greeting[0] */

/* Print out the greeting string */
printf("%s\n", pCharArray);

*pCharArray = 'H';        /* Change greeting[0] to 'H' */
*(pCharArray + 1) = 'i';  /* Change greeting[1] to 'i' */
pCharArray[2] = '\0';     /* Change greeting[2] to '\0' */

/* Print out the new string */
printf("%s", pCharArray);
```

Sample Run,

```
z:\apc-cc-conduit\images>a
Hello World.
Hi
z:\apc-cc-conduit\images>
```

BE CAREFUL ABOUT USING POINTERS THAT DO NOT POINT TO MEMORY ALLOCATED FOR YOUR APPLICATION. RESULTS CAN BE UNPREDICTABLE. FOR EXAMPLE, THE FOLLOWING CODE PERFORMS A STRING COPY TO A GARBAGE MEMORY LOCATION POINTED TO BY pChar. THE BEHAVIOR IS UNPREDICTABLE.

```
#include <stdio.h>
#include <string.h>

int main(void)
{
      char *pChar;
      char *pMsg;
      char message[] = "Hello World.";

      pMsg = message;

      strcpy(pChar, pMsg);

      printf("\n%s\n", pChar);

      return 0;
}
```

```
/* Creating literal strings and printing them out */
pCharArray = "I like to program."; /* Points to literal string */
printf("%s\n", pCharArray);

pCharArray = "Do you like to program?  I hope so.";
printf("%s", pCharArray);
```

Sample Run,

```
z:\apc-cc-conduit\images>a
I like to program.
Do you like to program?  I hope so.
z:\apc-cc-conduit\images>
```