1. Write, compile and run the below example demonstration program. 
Reference: http://pw1.netcom.com/~tjensen/ptr/ch3x.htm

```c
#include <stdio.h>

char strA[80] = "A string to be used for demonstration purposes";
char strB[80];

int main(void)
{
    char *pA;     /* a pointer to type character */
    char *pB;     /* another pointer to type character */
    puts(strA);   /* show string A */
    pA = strA;    /* point pA at string A */
    puts(pA);     /* show what pA is pointing to */
    pB = strB;    /* point pB at string B */
    putchar('\n');       /* move down one line on the screen */
    while(*pA != '\0')   /* line A (see text) */
    {
        *pB++ = *pA++;   /* line B (see text) */
    }
    *pB = '\0';          /* line C (see text) */
    puts(strB);          /* show strB on screen */
    return 0;
}
```

2. Define the below functions: (HINT: Below functions and their usages are similar to “strcpy()” and “strncpy()” that are available at “string.h”.)

```c
char *my_strcpyF1(char *destination, const char *source);
char *my_strcpyF2(char destination[], char source[]);
char *my_strncpy(char *destination, const char *source, size_t n);
```

3. Write comments for the dynamic memory allocation method used within the below program. 
Reference: http://pw1.netcom.com/~tjensen/ptr/ch9x.htm

```c
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int nrows = 5;     /* Both nrows and ncols could be evaluated */
    int ncols = 10;    /* or read in at run time */
    int row,col;
    int **rowptr;

    rowptr = (int**)malloc(nrows * sizeof(int *));
    if (rowptr == NULL)
    {
        puts("\nFailure to allocate room for row pointers.\n");
        exit(0);
    }
```
for (row = 0; row < nrows; row++)
{
    rowptr[row] = (int*) malloc(ncols * sizeof(int));
    if (rowptr[row] == NULL)
    {
        printf("\nFailure to allocate for row[%d]\n",row);
        exit(0);
    }
}

printf("\n\n\n");

for (row = 0; row < nrows; row++)
{
    for (col = 0; col < ncols; col++)
        rowptr[row][col]=2*row*col;
}

for (row = 0; row < nrows; row++)
{
    for (col = 0; col < ncols; col++)
        printf("%d	",rowptr[row][col]);
    printf("\n");
}

return 0;