HW4

There is a bug in the strcpy.c code.

```
/*

  File: strcpy.c

  Description: Copies a source string to a destination. Keeps copying
  until it finds the NULL char in the source char string

  Input: char pointers for source (s2) and destination (s1)

  Output: returns the pointer to the destinatio (s1)

*/

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

char *my_strcpy(char * , const char * );

int main()
{

  char src[] = "cs23!";
  char dst[]="Hello hello";
  char *curdst;
  int len=0;

  printf("src address %p and first char %c \n", (void *)&src, src[0]);
  printf("dst address %p and first char %c \n", (void *)&dst, dst[0]);

  // compute where NULL character is '\0' ASCII 0

  while(src[len++]);

  // print out the char arrays and various addresses.

  printf("src array %s and last element %d\n", src, atoi(&src[len]));
  printf("dst array %s and last element %c\n", dst, dst[len]);

  // do the copy

  curdst= my_strcpy(dst, src);

  // check to see if the NULL char is copied too.

  printf("dst array %s and last element %d\n", dst, atoi(&dst[len]));

  return 0;

}

char *my_strcpy(char *s1, const char *s2) {

  register char *d = s1;

  // print the pointer variables address and their contents, and first char

  printf("s2 address %p, its contents is a pointer %p to first char %c \n", (void *)&s2, (void *)s2, *s2);
  printf("s1 address %p, its contents is a pointer %p to first char %c \n", (void *)&s1, (void *)s1, *s1);

  while ((*d++ = *s2++));
  return(s1);

}
```

As we can see there's a code that make two different character-type arrays copied. To copy the array. We cannot use the normal way as variable does. We should use the pointer to point the arrays addresses. And the code up there show how to do a string-copy using pointer. But there's a bug in the code. At the line of "While(src[len++])" , we know if you want the while loop can run successfully.

You need to set a condition. However, the while loop's condition src[len] will cause a problem that the src array's length doesn't fit the dst array's length. So, the code still can run, but it turns out result wrong.

```
src array cs23! and last element 0
dst array Hello hello and last element h
```

The last element of dst array is not h.