String Functions

Last updated 10/29/20

String Functions

- These slides discuss several string functions
- Upon completion: You should be able interpret and code using string functions

String Functions

- String Functions
  - printf()

```
printf(“my string is: %s”, myString);
```

Note: printf allows strings to be printed by name – no need to cycle through the elements

  - scanf()

```
char month[10];                  // create string
scanf(“%9s”, month);             // read in 9 characters
                                 // for the string called
                                 // month (adds the \0)


fflush(stdin);                   // required to remove
                                 // any extra characters
                                 // and the newline
```

** if we read in more characters than the string can hold we will overwrite unrelated data – don’t forget 1 is used for the terminator

String Functions

- String Functions
  - Get string – converts a line (up to newline) to a string
  - gets(char* stringPtr)

```
char myString[81];          // standard 80 character line
                            // must be big enough to hold
                            // your line
…
gets(myString);             // read one line of input
```

String Functions

- String Functions
  - Put string – converts a string to a line of output (including the newline)
  - puts(const char* stringPtr)

```
…
puts(myString);                 // output 1 line with value
                                // myString
```

String Functions

- String Functions - #include <string.h>

  - String length – outputs the length of a string excluding the null character
  - int strlen(const char* string)

```
…
foo = strlen(myString);
```

String Functions

- String Functions - #include <string.h>

  - String copy – copy one string to another
  - char* strcopy(char* toStr, const char* fromStr)
    - returns the address of toStr

```
…
strcopy(string2, string1);
```

NO Boundary or Size checking is done

  - Use strncopy
  - char* strncopy(char* toStr, const char* fromStr, int size)

String Functions

- String Functions - #include <string.h>

  - String compare – compare 2 strings
  - int strcmp(const char* str1, const char* str2)
    - returns 0 if equal
    - returns <0 if str1 < str2
    - returns >0 if str1 > str2

Note: compares ascii values

```
if(strcmp(mystr1, mystr2) == 0)
   …
```

  - int strncmp(const char* str1, const char* str2, int size)
  - Compares the first N elements

String Functions

- String Functions - #include <string.h>

  - String concatenation – concatenate 2 strings
  - char* strcat(char* str1, const char* str2)
    - Returns the address of str1
    - Concatenates str2 onto str1 with result in str1

```
...
strcat(stringA, stringB);     // result in stringA
```

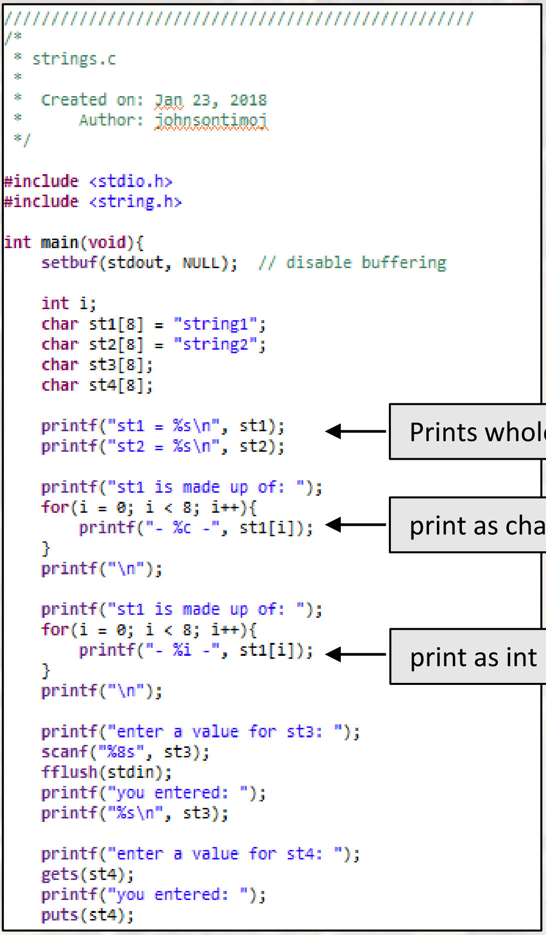
NO Boundary or Size checking is done

  - Use strncat
  - char* strncat(char* str1, const char* str2, int size)

String Functions

- String Functions

- #include <string.h>

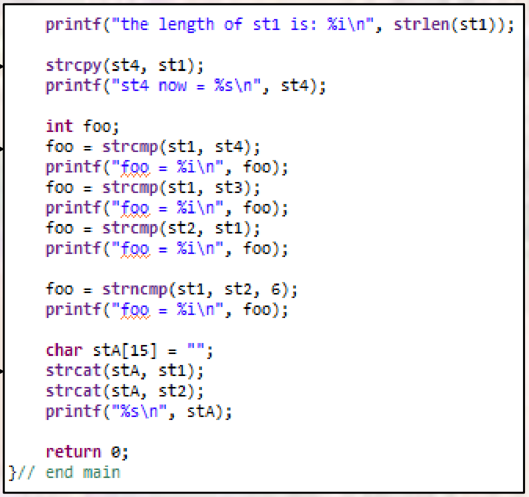


```
//////////////////////////////////////////////////////
/*
 * strings.c
 *
 *  Created on: Jan 23, 2018
 *      Author: johnsontimoj
 */

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

int main(void){
    setbuf(stdout, NULL);  // disable buffering

    int i;
    char st1[8] = "string1";
    char st2[8] = "string2";
    char st3[8];
    char st4[8];

    printf("st1 = %s\n", st1);
    printf("st2 = %s\n", st2);

    printf("st1 is made up of: ");
    for(i = 0; i < 8; i++){
        printf("- %c -", st1[i]);
    }
    printf("\n");

    printf("st1 is made up of: ");
    for(i = 0; i < 8; i++){
        printf("- %i -", st1[i]);
    }
    printf("\n");

    printf("enter a value for st3: ");
    scanf("%8s", st3);
    fflush(stdin);
    printf("you entered: ");
    printf("%s\n", st3);

    printf("enter a value for st4: ");
    gets(st4);
    printf("you entered: ");
    puts(st4);
```

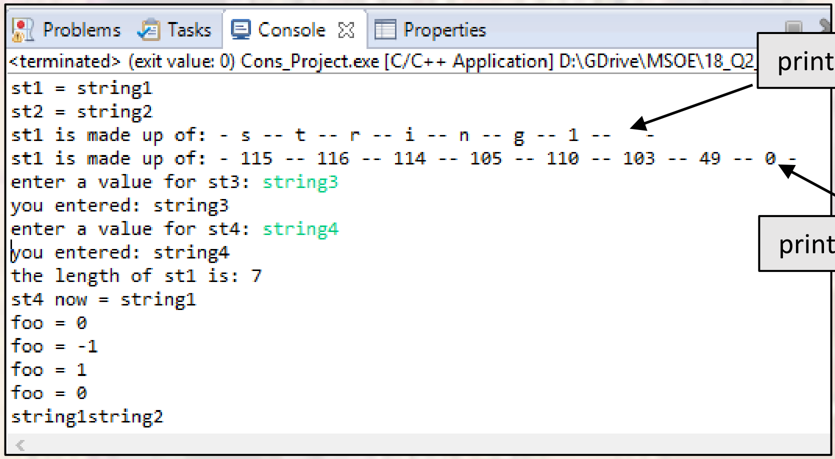
Prints whole string

print as char

print as int



```
    printf("the length of st1 is: %i\n", strlen(st1));

    strcpy(st4, st1);
    printf("st4 now = %s\n", st4);

    int foo;
    foo = strcmp(st1, st4);
    printf("foo = %i\n", foo);
    foo = strcmp(st1, st3);
    printf("foo = %i\n", foo);
    foo = strcmp(st2, st1);
    printf("foo = %i\n", foo);

    foo = strncmp(st1, st2, 6);
    printf("foo = %i\n", foo);

    char stA[15] = "";
    strcat(stA, st1);
    strcat(stA, st2);
    printf("%s\n", stA);

    return 0;
}// end main
```



```
<terminated> (exit value: 0) Cons_Project.exe [C/C++ Application] D:\GDrive\MSOE\18_Q2
st1 = string1
st2 = string2
st1 is made up of: - s -- t -- r -- i -- n -- g -- 1 --  -
st1 is made up of: - 115 -- 116 -- 114 -- 105 -- 110 -- 103 -- 49 -- 0 -
enter a value for st3: string3
you entered: string3
enter a value for st4: string4
you entered: string4
the length of st1 is: 7
st4 now = string1
foo = 0
foo = -1
foo = 1
foo = 0
string1string2
```

print terminator as char

print terminator as int