Esc101: Strings

Instructor: Krithika Venkataramani

Semester 2, 2011-2012

The contents of most of these slides are from the lecture slides of Prof. Arnab Bhattacharya

Array of characters: String

- A string is an array of characters
- Strings are not supported as separate data types
- A string can be declared in the following way:
  - char name [30];
- Strings can be initialized in double quotes:
  - char city [ ] = "Kanpur";
- Strings can be initialized as array of characters:
  - char ct [ ] = {'K', 'a', 'n', 'p', 'u', 'r', '\0'};
- A string is an array of characters terminated with the null character '\0'
  - So, array city contains '\0' in the end and its size is 6 + 1 = 7

String input and output

- A string can be read/written using "%s" specification
  - char name [30];
  - scanf ("%s", name);
    - there is no "&" before the string variable name
  - printf ("%s", name);
- Reading will stop as soon as the first whitespace character (blank, tab, return) is encountered
- If size of array is larger, rest of the characters are undefined
- If size of array is smaller, reading overlaps beyond the array boundary
- A string is printed up to the null character

Reading and displaying strings

```
# include <stdio .h>
int main ()
{
char city [ ] = "Kanpur";
char ct [ ] = {'D', 'e', '\n', 'l', 'h', 'i', '\0'};
char sr [6] = "India";
char sm [7] = "Asia";
char sl [4] = "World";
char str1 [4] , str2 [4] , str3 [4];
printf ("%s\n", city );
```


Reading and displaying strings (cont.)

```
printf ("%s\n", ct);
printf ("%s\n", sr);
printf ("%s\n", sm);
printf ("%s\n", sl);
scanf ("%s", str1 );
scanf ("%s", str2 );
scanf ("%s", str3 );
printf ("%s\n", str1 );
printf ("%s\n", str2 );
printf ("%s\n", str3 );
}
```


Reading/writing strings with blank spaces

- To read a string with blanks and tabs, use gets
- char name [30];
- gets(name);
- gets function does not perform boundary checks and may thus overflow and write characters in memory locations not intended
- puts() is the display equivalent of gets()
- puts automatically appends a return character
- puts(name);

String Operations

- Include the library string.h
- strlen(s) returns the length of string s
- It does not count the null character
- strcpy(duplicate, original) copies string original to duplicate
- strcat(s, t) appends string t to s
- strcmp(s, t) returns the lexicographic comparison between strings s and t
  - If they are equal, 0 is returned
  - If s is later than t, a positive integer is returned
  - If s is earlier than t, a negative integer is returned
  - Capital letters come earlier than small letters
  - No-letter comes earlier than any other letter
  - Blank comes earlier than other letters

String operations

```
# include <stdio .h>
# include <string .h>
int main ()
{
    char str [] = " Kanpur ";
    char ca [] = {'K', 'O', 'L', 'K', 'A', 'T', 'A', '\0 '};
    char extra [30] = "e";
    printf ("%d\t%d\n", strlen (str), strlen (ca));
    printf ("%s\n", extra );
    strcpy (extra , ca);
    printf ("%s\n", extra );
```


String operations (cont.)

```
    strcat (extra , str );
    printf ("%s\n", extra );
    printf ("%d\t%d\t%d\n", strcmp (str , ca), strcmp (ca , extra ),
    strcmp (extra , str ));
    printf ("%d\n", strcmp ("a b", "ab"));
}
```


String copy using arrays

- strcpy(s, t) copies string t to s
- Array version

```
int i = 0;
char s[30], t[30];
printf("Enter the string t:");
scanf("%s",t);
while (t[i] != '\0 ') // t has not finished
{
    s[i] = t[i]; // copy
    i++;
}
s[i] = '\0 '; // terminate s
```


String compare using arrays

- strcmp(s, t) returns the comparison between strings s and t
- Array version

```
int i;
for (i = 0; s[i] == t[i]; i++) // traverse equal elements
    if (s[i] == '\0 ') // t[i] is also '\0'
        printf("0: same strings"); // only equal elements
```

Review of scanf

Declaration	Statement	Input (□ means blank)	Value stored
char str[8]	scanf("%s",str);	□\thello\n	hello\0
		□New□York\n	New\0
		Exceeding□\n	Exceeding\0

Placeholders with printf

Value	Placeholder	Output (□ means blank)	Interpretation
"fantastic"	%s	fantastic	print string
	%6s	fantastic	print string in 6 columns. If string is greater than 6, entire string is printed
	%12s	□□□fantastic	print string in 12 columns.
	%-12s	fantastic□□□	print string left-justified in 12 columns
	%4.3s	□fan	print first 3 characters in 4 columns
	%.3s	fan	print first 3 characters

Sample program

- Read in a long string, with blanks and tabs, till the end of a line. Display one additional character in each successive line till the whole line is displayed.
  - Input: ESc101- Fundamentals of Computing
  - Output is as follows
  - E
  - ES
  - ESc
  - ESc1
  - --
  - ESc101- Fundamentals of Computing

Formatted output: Version 1 using nested loops

```
printf("Enter the string\n");
gets(str); //read the string using gets
for (count=0; str[count]!='\0'; count++)
{//count increases in each iteration till the null character
//loop to print an additional character in each successive line
    for(j=0; j<=count; j++)
        printf("%c",str[j]);
    printf("\n"); //new line
}
```

Version 2 using string formatting with printf

```
/*Left justified triangle like previous*/
    for (count=0; str[count]!='\0'; count++)
        printf("\n%.*s",count,str);  /*display the first 'j+1' characters of the
    string 'str'*/
/*To print a right justified triangle like below
                                    E
                                   ES
                                  ...
ESc101:Fundamentals of Computing*/
    for (count=0; str[count]!='\0'; count++)
    { } //count= number of characters in the string, other than '\0'
    for (j=0; j<count; j++)
        printf("\n%*.*s",count,j+1,str); /*displays first j+1 characters, right
    justified in columns equal to entire string length*/
```


Converting from String to Number

- Can convert string having numbers to integer/float format
  - Example: Read date in dd:mm:yyyy format using a string
  - Convert the digits corresponding to dd, mm, yyyy to integer
- Include library stdlib.h
- x=atoi(string) converts from character string to integer, x.
- x=atof(string) converts from character string to float, x.
- Example: conversion of date dd:mm:yyyy from character sequence to integers
- Example: conversion of pin code in address string to an integer

Conversion from string to numbers

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int day, month, year;
    char dd[3],mm[8],yyyy[8],stringf[10];
    float f;
    printf("Float:");
    scanf("%s",stringf);
    f = atof(stringf);  //conversion to a float
    printf("Number = %f\n",f);
    printf("Enter the date in dd:mm:yyyy format: ");
    scanf(" %c%c:%c%c:%c%c%c%c",&dd[0],&dd[1],&mm[0],&mm[1],&yyyy[0],
    &yyyy[1],&yyyy[2],&yyyy[3]); //read date in string format
    /*scanf expects : and does not store it in the characters*/
```


Conversion from string to numbers (cont.)

```
    dd[2] = '\0'; //dd is a string storing the sequence of numerals having the day
    mm[2] = '\0'; //mm is a string storing the numerals of the month
    yyyy[4]='\0'; //mm is a string storing the numerals of the year
    printf("Strings:dd=%s, mm=%s, yyyy=%s",dd,mm,yyyy);
    day = atoi(dd);  //conversion to integer day
    month = atoi(mm); //conversion to integer month
    year = atoi(yyyy); //conversion to integer year
    printf("Date: %d/%d/%d\n",day,month, year);
}
```

Example on strings

- Write a program that accepts the name of the country as input and displays the corresponding capital. The program should continue to take inputs until 'End' is entered. Consider the following countries.
- India – New Delhi
- Italy – Rome
- China – Beijing
- Sri Lanka – Colombo
- Japan – Tokyo
- United States of America – Washington D.C.

Algorithm to display country capitals

1. Input country. Read characters till end of line to create a string.
2. If country is India, display "New Delhi".
3. Otherwise, if country is Italy, display "Rome".
4. --
5. If country is not "End", go back to Step 1.

- Points to remember while coding
- string has to end with a '\0'
- string comparison requires a character by character comparison
- strcmp() – function for string comparison returns 0 for exactly equal strings

Code to display country capitals

```
#include<stdio.h>
#include<string.h>
void main()
{
    char country[100];
    int j;
    gets(country);
    do
    {
        /*Printing Capitals of Countries*/
    } while( strcmp(country,"End") !=0);
}
```


Code within do-while loop to display country capitals

```
/*Printing Capitals of Countries*/
if( strcmp(county,"India")==0 )
    printf("New Delhi\n");
else if ( strcmp(county,"Sri Lanka")==0 )
    printf("Colombo\n");
//-- add other country capitals
else if(country,"End")
    break;
else
    printf("Country capital not known\n");
printf("Enter a new country");
gets(country);
```