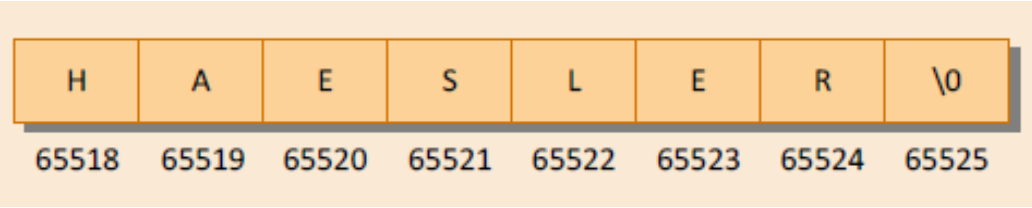
Strings

IN 1101 PROGRAMMING FUNDAMENTALS

What are Strings?

- Group of characters can be stored in a character array.
- Many languages internally treat ‘strings’ as character arrays.
- The array of characters are arranged one after another in memory as successive elements.
- A string is always terminated by a null character (i.e. slash zero \0).

Example: char name[ ] = { 'H', 'A', 'E', 'S', 'L', 'E', 'R', '\0' } ;



- Note : ‘\0’ is not necessary in some declarations.
  e.g.: char name[ ] = "HAESLER" ;
- The **%s** used in printf( ) is a format specification for printing out a string.

Example:

```
# include <stdio.h>
int main( )
{
   char name[ 25 ] ;
   printf ( "Enter your name " ) ;
   scanf ( "%s", name ) ;
   printf ( "Hello %s!\n", name ) ;
   return 0 ;
}
```

Using gets() and puts()

- scanf( ) is not capable of receiving multi-word strings.
- The way to get around this limitation is by using the function gets( ) and its counterpart puts( ).
- puts( ) can display only one string at a time.
- Unlike printf( ), puts( ) places the cursor on the next line.

Example: gets() and puts()

```
# include <stdio.h>
int main( )
{
    char name[ 25 ] ;
    printf ( "Enter your full name: " ) ;
    gets ( name ) ;
    puts ( "Hello!" ) ;
    puts ( name ) ;
    return 0 ;
}
```

Function	Use
strlen	Finds length of a string
strlwr	Converts a string to lowercase
strupr	Converts a string to uppercase
strcat	Appends one string at the end of another
strncat	Appends first n characters of a string at the end of another
strcpy	Copies a string into another
strncpy	Copies first n characters of one string into another
strcmp	Compares two strings
strncmp	Compares first n characters of two strings
strcmpi	Compares two strings by ignoring the case
stricmp	Compares two strings without regard to case (identical to strcmpi)
strnicmp	Compares first n characters of two strings without regard to case
strdup	Duplicates a string
strchr	Finds first occurrence of a given character in a string
strrchr	Finds last occurrence of a given character in a string
strstr	Finds first occurrence of a given string in another string
strset	Sets all characters of string to a given character
strnset	Sets first n characters of a string to a given character
strrev	Reverses string

Standard String Functions

- Need to include “string.h” for using string functions.

strlen( )

- Counts the number of characters present in a string.

```
# include <stdio.h>
# include <string.h>
int main( )
{
   char arr[ ] = "Programming Fundamentals" ;
   int len1, len2 ;
   len1 = strlen ( arr ) ;
   len2 = strlen ( "University of Moratuwa" ) ;
   printf ( "string = %s length = %d\n", arr, len1 ) ;
   printf ( "string = %s length = %d\n", "University of Moratuwa", len2 ) ;
   return 0 ;
}
```

```
-------Output--------

string = Programming Fundamentals length = 24
string = University of Moratuwa length = 22
```

strcpy( )

- Copies the contents of one string into another.

```
# include <stdio.h>
# include <string.h>
int main( )
{
   char source[ ] = "Morning" ;
   char target[ 20 ] ;
   strcpy ( target, source ) ;
   printf ( "source string = %s\n", source ) ;
   printf ( "target string = %s\n", target ) ;
   return 0 ;
}
```

```
-------Output--------

source string = Morning
target string = Morning
```

strcat( )

- Concatenates the source string at the end of the target string.

```
# include <stdio.h>
# include <string.h>
int main( )
{
   char source[ ] = "Morning!" ;
   char target[ 30 ] = "Good" ;
   strcat ( target, source ) ;
   printf ( "source string = %s\n", source ) ;
   printf ( "target string = %s\n", target ) ;
   return 0 ;
}
```

```
------Output--------

source string = Morning!
target string = GoodMorning!
```

strcmp( )

- This is a function which compares two strings to find out whether they are same or different.
- The two strings are compared character-by character until there is a mismatch or end of one of the strings is reached, whichever occurs first.
- Returns value of 0 if s1 the same as s2, returns -ve value if s1 less than s2, and returns +ve value if s1more than s2.

strcmp( ) Cont.

```
# include <stdio.h>
# include <string.h>
int main( )
{
   char string1[ ] = "Jerry" ;
   char string2[ ] = "Ferry" ;
   int i, j, k ;
   i = strcmp ( string1, "Jerry" ) ;
   j = strcmp ( string1, string2 ) ;
   k = strcmp ( string1, "Toy" ) ;
   printf ( "%d %d %d\n", i, j, k ) ;
   return 0 ;
}
```

```
-------Output-------

0 1 -1
```

Exercise

Write a program that extracts part of the given string from the specified position. For example, if the sting is "Working with strings is fun", then if from position 4, 4 characters are to be extracted

then the program should return string as "king". If the number of characters to be extracted is 0 then the program should extract entire string from the specified position.

Questions?