C Programming Introduction

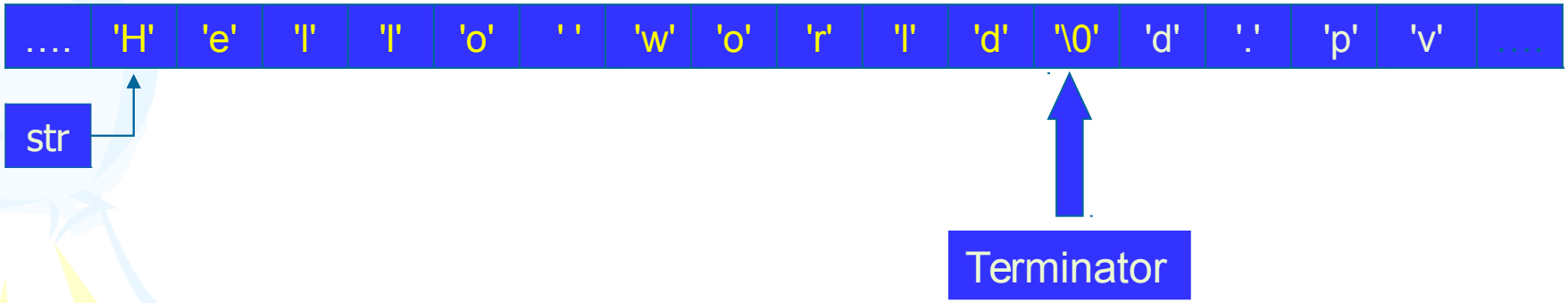
week 13: Strings

Strings

- An array of characters
- Used to store text
- Another way to initialize:

```
char str[] = "Text";
```

The terminator



The Terminator

- Strings terminate with NULL character, signed by '\0' (ascii code 0)
- This is a convention used to know where the string ends
- It means that in order to hold a string of N characters we need an array of length N + 1
- So the previous initialization is equivalent to

```
char str[] = {'b', 'l', 'a', 'b', 'l', 'a', '\0'};
```

String library

- Like in the case of stdio.h and math.h, we have a special library for handling strings
- We should **#include <string.h>**

String library

- Functions:
  - **`strlen(const char s[])`**
    returns the length of s
  - **`strcmp(const char s1[],`**
    `const char s2[])`
    compares s1 with s2
  - **`strcpy(char s1[],`**
    `const char s2[])`
    copies to contents of s2 to s1
  - and more…

Example

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[ 20 ] = "Happy ";
    char s2[] = "New Year ";
    char s3[ 40 ] = "";
    printf( "s1 = %s\ns2 = %s\n", s1, s2 );
    printf( "strcat( s1, s2 ) = %s\n", strcat( s1, s2 ) );
    printf( "strncat( s3, s1, 6 ) = %s\n", strncat( s3, s1, 6 ) );
    printf( "strcat( s3, s1 ) = %s\n", strcat( s3, s1 ) );
    return 0;
}
```

```
s1 = Happy
s2 = New Year
strcat( s1, s2 ) = Happy New Year
strncat( s3, s1, 6 ) = Happy
strcat( s3, s1 ) = Happy Happy New Year
```

String Conversion Functions

- Conversion functions
  - In **<stdlib.h>** (general utilities library)
- Convert strings of digits to integer and floating-point values

Prototype	Description
`double atof( const char *nPtr )`	Converts the string **nPtr** to **double**.
`int atoi( const char *nPtr )`	Converts the string **nPtr** to **int**.
`long atol( const char *nPtr )`	Converts the string **nPtr** to long **int**.

Character Analysis and Conversion

Functions (ctype.h)	Description
`isalpha`	Check if the argument is a letter
`isdigit`	Check if the argument is one of the ten digits
`isspace`	Check if argument is a space, newline or tab.
`tolower`	Converts the lowercase letters in the argument to upper case letters.

String conversion function

Prototype	Description
`double atof( const char *nPtr )`	Converts the string **`nPtr`** to **`double`**.
`int atoi( const char *nPtr )`	Converts the string **`nPtr`** to **`int`**.
`long atol( const char *nPtr )`	Converts the string **`nPtr`** to long **`int`**.

Arrays of Strings

- An array of strings is a two-dimensional array of characters in which each row is one string.
  - `char names[People][Length];`
  - `char month[5][10] = {“January”, “February”, “March”, “April”, “May”};`

Exercise 13.1

- Write a program that inputs a line of text, counts the number of blanks by using a function, and displays the number of blanks.

Solution

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

void spacecounter(char []);   // Function
  prototype

 void main(void)
{
  char line[81];
  printf("Enter a line of text:\n");
  gets(line);
  printf("Blanc character occurs for: %d time
  in the line.\n", spacecounter(line));
}
```

Solution

```
int spacecounter(char inputline[])
{
   int i = 0;
   int count = 0;
   while (inputline[i] != '\0') {
      if (inputline[i] == ' ')
           count++;
      }
      i++;
   }
   return count
}
```

Exercise 13.2

- write a function that:
  - gets a string and two chars
  - the functions scans the string and replaces every occurrence of the first char with the second one.
- write a program to test the above function
  - the program should read a string from the user (no spaces) and two characters, then call the function with the input, and print the result.
- example
  - input: “papa”, ‘p’, ‘m’
  - output: “mama”

Solution (function)

```
void replace(char str[], char replace_what,
                                 char replace_with)
{
    int i;

    for (i = 0; str[i] != '\0'; ++i)
    {
        if (str[i] == replace_what)
            str[i] = replace_with;
    }
}
```

Solution (main program)

```
#define STRING_LEN 100

int main(void)
{
    char str[STRING_LEN + 1];
    char replace_what, replace_with;

    printf("Please enter a string (no spaces)\n");
    scanf("%100s", str);

    printf("Letter to replace: ");
    scanf(" %c", &replace_what);

    printf("Letter to replace with: ");
    scanf(" %c", &replace_with);

    replace(str, replace_what, replace_with);

    printf("The result: %s\n", str);

    return 0;
}
```

Exercise 13.3

- Write a program that tests a customer number to determine whether it is in the proper format(LLLNNNN with LLL are letters and NNNN are numbers).

Solution: testNum function

```
bool testNum(char custNum[])
{
   // Test the first three characters for alphabetic
   letters
   for (int count = 0; count < 3; count++)
   {
       if (!isalpha(custNum[count]))
              return false;
   }
   // Test the last 4 characters for numeric digits
   for (int count = 3; count < 7; count++)
   {
       if (!isdigit(custNum[count]))
              return false;
   }
   return true;
}
```

Solution: main program

```
#include <stdio.h>
#include <ctype.h>

bool testNum(char []);

void main(void)
{
   char customer[8];
   printf("Enter a customer number with exact 7 characters
   in the form LLLNNNN\n";
   printf("LLL = letters and NNNN = numbers): ";
   scanf("%s",customer);
   if (testNum(customer))
        printf("That's a valid customer number.\n";
   else
   {
       printf("That is not the proper format of the
   customer   number.\nHere is an example: ABC1234\n");
   }
}
```

Exercise 13.3

- Write your own replacement for the standard **strcpy()** that comes with C without using string.h

Solution

```
char *my_strcpy(char *destination, char *source)
  {
     char *p = destination;
     while (*source != '\0')
     {
        *p++ = *source++;
     }
     *p = '\0';
     return destination;
  }
```

Exercise 13.4

- Write a program asks the user to enter his or her first and last names, separated by a space. Then print out the first name.
- The program shoud use a function which cuts off the last name off the string in parameter.

Solution

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

void nameSlice(char []);       // Function prototype

 void main(void)
{
   char name[41];
   printf("Enter your first and last names, separated
   by a space:\n");
   gets(name);
   nameSlice(name);
   printf("Your first name is: %s\n", name);
}
```

Solution

```
// This function accepts a character array as its
// argument. It scans the array looking
// for a space. When it finds one, it replaces it
// with a null terminator.

void nameSlice(char userName[])
{
   int count = 0;
   while (userName[count] != ' ' &&
   userName[count] != '\0')
      count++;
   if (userName[count] == ' ')
      userName[count] = '\0';
}
```

Exercise 13.5

- Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

Solution: strend function

```
#include <string.h>
int strend(char *s, char *t)
{
   int Result = 0;
   int s_length = 0;
   int t_length = 0;

   /* get the lengths of the strings */
   s_length = strlen(s);
   t_length = strlen(t);

   if(t_length <= s_length)  {
   /* advance the s pointer to where the string t would have to start in string s */
    s += s_length - t_length;
        /* and make the compare using strcmp */
        if(0 == strcmp(s, t))    {
        Result = 1;
        }
   }
  return Result;
}
```

Solution: main program

```
#include <stdio.h>
int main(void)
{
  char *s1 = "some really long string.";
  char *s2 = "ng.";
  char *s3 = "ng";
  if(strend(s1, s2)) {
    printf("The string (%s) has (%s) at the end.\n", s1, s2);
  } else {
    printf("The string (%s) doesn't have (%s) at the end.\n", s1, s2);
  }
  if(strend(s1, s3)){
    printf("The string (%s) has (%s) at the end.\n", s1, s3);
  }
  else {
    printf("The string (%s) doesn't have (%s) at the end.\n", s1, s3);
  }
  return 0;
}
```

Exercise 13.6: Using strstr

- A list of product number and description of shop is:

```
"TV127  31 inch Television",
"CD057  CD Player",
"TA877  Answering Machine",
"CS409  Car Stereo",
"PC655  Personal Computer"
```

- Store this list in an array of string and write a program allowing user to lookup a product description by entering all or part of its product number.

Solution

```
#include <stdio.h>
#include <string.h> // For strstr

void main(void)
{
   char prods[5][27] = {"TV127  31 inch Television",
                        "CD057  CD Player",
                        "TA877  Answering Machine",
                        "CS409  Car Stereo",
                        "PC655  Personal Computer"};
   char lookUp[27], *strPtr = NULL;
   int index;
   printf("\tProduct Database\n\n");
   printf("Enter a product number to search for: ");
   scanf("%s",lookUp);
```

Solution

```
for (index = 0; index < 5; index++)
{
    strPtr = strstr(prods[index], lookUp);
    if (strPtr != NULL)
    break;
}
if (strPtr == NULL)
     printf("No matching product was
          found.\n");
else
     printf("%s\n", prods[index]);
}
```

Exercise 13.7

- Write a program that accepts a string from the user
  and replaces all punctuation signs (`,.;:!?`) with spaces

Solution (str_any.c)

```
char* str_any(char* str1, char* str2)
{
    while (*str1 != '\0')
    {
        if (strchr(str2, *str1) != NULL) {
            return str1;
        }
        ++str1;
    }

    return NULL;
}
```

Solution

```
int main(void)
{
    char* punc = ".,;:!?";
    char s[MAX_LENGTH + 1];
    char *p;

    printf("Please enter a line of text\n");
    scanf("%100s", s);

    for (p = str_any(s, punc);
         p != NULL;
         p = str_any(p + 1, punc)) {
        *p = ' ';
    }

    printf("Resulting string is:\n%s\n", s);

    return 0;
}
```