C Programming

Strings

Converting numeric strings to numbers

If you had the following string:

```
char num_string[4] = "123";
```

The above string consists of 3 individual characters, i.e., '1', '2', and '3' as well as the NULL character at the end.

Combined, the string is not a numeral, i.e., it is different from the integer value 123.

There will be times when you will be using strings representing numeric values but need to convert these into their equivalent numeric value for use in some arithmetic operation. You can write your own code to do this, however C provides built-in functions for this conversion process.

The built-in functions for this conversion process is provided by:

#include <stdlib.h>

Here is sample code that shows the conversion process:

```
/*
Converting numeric strings into the equivalent numeric value
*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
  char num_string[4] = "123";
  int my_num = 0;

  // atoi() means ascii-to-integer. This converts a numeric
string into its
```

```
  // equivalent integer

  my_num = atoi(num_string);

  printf("\nmy_num contains %d", my_num);

  return 0;

} // end main()
```

Repl 19.1: https://replit.com/@michaelTUDublin/191-Convert-numeric-string-to-a-number

Also, you can convert a numeric string into a float using **atof()** as follows:

```
/*
Converting numeric strings into the equivalent numeric value
*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
  char num_string[4] = "124";
  float my_num = 0;

  // atof() means ascii-to-float. This converts a numeric string
into its equivalent float
  my_num = atof(num_string);

  printf("\nmy_num contains %f", my_num);

  return 0;

} // end main()
```

Arrays of strings

We know that a string is an array of characters.

Therefore, what do you do if you require an array of strings?

In order to create an array of strings, this can be done by creating an array of pointers !!!
This may seem strange but think about one of the ways you can initialise a single string, i.e.,

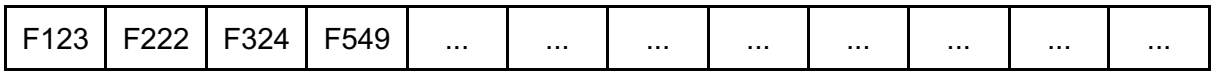
```
char *str = "Hello";
```

By using an array of character pointers, each element in the array will contain a separate memory address, which is the memory address of the first character of different strings.

e.g.,

```
char *months[12] = {"Jan", "February", "March", "Apr",
                    "May", "Jun", "Jul", "Aug",
                    "Sep", "Oct", "Nov", "Dec"};
```

months



0 1 2 3 4 5 6 7 8 9 10 11

array element number

```
/*

Program that uses an array of strings, i.e., it specifically uses
an array of pointers, each pointer will point to the memory
address of the first character of that string

*/

#include <stdio.h>
```

```c
#define SIZE 12

int main()

{

   char *months[SIZE] = {"Jan", "February", "March", "April",

                         "May", "June", "July", "Aug",

                         "Sep", "Oct", "Nov", "December"};

   int i;


   /* Display each string contained in the array to standard
output

   */

   printf("\nThe months of the year are:\n");


   for(i = 0; i < SIZE; i++)

   {

       printf("%s\n", months[i]);

   } // end for

   return 0;

} // end main()
```

Repl 19.2: https://replit.com/@michaelTUDublin/192-Array-of-strings