

Strings

One common technique used to represent string variables is to let these variables be pointers into a single common character array which is used to store the actual characters for the strings. In C, we can define the type `STRNG`, the character array `CHARTEXT`, and the string space `STRING_SPACE` as follows:

```
#ifndef STR_H_
#define STR_H_

#define MAX_STRING_SPACE 1024
#define MAX_STRING_SIZE 64
#define MAX_NUMBER_OF_STRINGS 16

typedef struct STRNG
{
    int ptr;
    int len;
} STRNG;

typedef struct STRING_SPACE
{
    char SPACE [MAX_STRING_SPACE];
    int nextAvailableSpace;

} STRING_SPACE;

typedef char *CHARTEXT;

void initstringspace (void);
void printstring (STRNG s);
void initstring( STRNG *s, CHARTEXT t, char delim);
void concat (STRNG s1, STRNG s2, STRNG *s3);

#endif
```

String variables are structures with two fields `ptr` (which is an index into `STRING_SPACE` and identifies the location of the first character in the string) and `len` (which is the string length). The variable `nextAvailableSpace` is used to indicate the next available location in `SPACE`.

Using this technique for representing string variables, implement the following procedures:

a) `void initstringspace (void);`

This function initializes all string space to be available.

b) `void printstring (STRNG s);`

This function will print out the string passed to the function.

c) `void initstring(STRNG *s, CHARTEXT t, char delim);`

This function creates a string variable `s` using the characters from `t`. All characters up to but not including the character found in `delim` are copied into string `s`.

d) `void concat (STRNG s1, STRNG s2, STRNG *s3);`

This function concatenates strings `s1` and `s2` and places the result in `s3`. (Note: `s1`, `s2`, and `s3` might not be different variables).

You are to write the code for each of the four functions listed above.

I will supply the driver code for this assignment via the Web by this evening, but here is some sample code that is to work with the above header file. Let's go over this for the sake of discussion.

```
#include <stdio.h>
#include <stdlib.h>
#include "str.h"

int main ()
{
    FILE *inFilePtr;
    int lineSize;
    CHARTEXT textLine = (CHARTEXT) malloc (MAX_STRING_SIZE);
    STRNG strings[MAX_NUMBER_OF_STRINGS];
    int numberOfStrings = 0;
    int i;

    if ((inFilePtr = fopen ("strings.txt", "r")) == NULL)
    {
        printf ("File Not Found \n");
        exit (EXIT_FAILURE);
    }
    else
    {
        while (fgets (textLine, MAX_STRING_SIZE, inFilePtr))
        {
            printf ("%s", textLine);
            initstring (&strings[numberOfStrings], textLine, '\0');
            printstring (strings[numberOfStrings]);
            printf ("\n");
            numberOfStrings++;
        }
    }

    for (i = 0; i < numberOfStrings; i++)
    {
        printf ("String #%d \n", i);
        printstring (strings[i]);
        printf ("\n");
    }

    return 0;
}
```