ΜΥΥ502 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

Εργαστήριο #4
3/11/2015

I. Array with Random Numbers

In this exercise you are aksed to implement a program where the user enter the size of an array and, then, the program dynamically allocates an array of that many elements which it fills with random numbers in the range [0, MAXNUM). Lastly, the program prints the contents of that array.

To generate random numbers you include the following headers:

```
#include <time.h>
#include <stdlib.h>
```

Moreover, somewhere in the beginning of the `main()` function, you must make the following function call:

```
srand(time(NULL));
```

Afterwards, you can generate random numbers from 0 to `MAXNUM-1` as:

```
num = rand() % MAXNUM;
```

II. Fill the voids...

You are given the program `lab4-2-fill.c` which is incomplete and you must fill in the spaces indicated by a comment '`FILL HERE`'. The purpose of that program is to ask from the user the size of an array, which it allocates dynamically and fills with some number (1 number). Then, it asks the user with some new array size and changes the size of the array making sure that any additional elements (in case the new size is larger) are assigned 0 (zero).

III. A glimpse from the midterm exam (last year)

Write a program in which the `main()` function receives arguments and does these things in order:

- Counts and sums each of the arguments length.
- Allocates memory of size equal to the above count (plus 1) in order to store a new string.
- Copies in the newly allocated space from the previous step the arguments (concatenation).
- Prints the new string.

IV. PASCAL Triangle

The famous *Pascal Triangle* is a triangluar provision of the binomial coefficients. For example, for $n = 5$ the pascal triangle is as:

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
```

The pascal triangle is a lower triangular matrix where the elements of the first column (column 0) and the elements of the main diagonal are equal to 1, whereas every other element in row i and column j is equal to the sum of the elements (row $i-1$ and column j) and (row $i-1$ and column $j-1$).

You must implement a function `int **pascal(int n);` that *dynamically* constructs Pascal's triangle with n rows and return it. Each row of the array must have space just to hold the elements required (i.e row i must have $i+1$ elements)

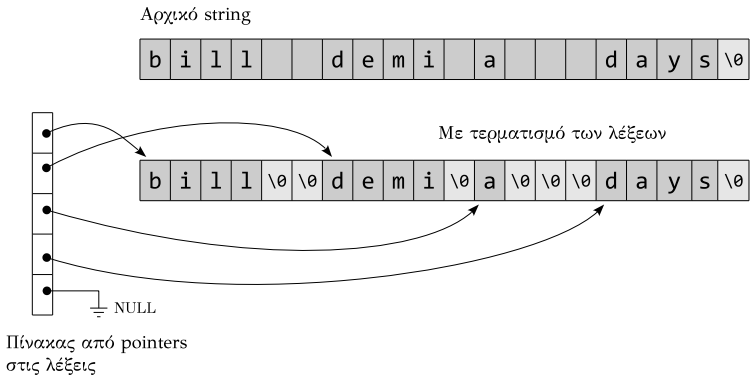
The `main()` function accepts as argument the parameter n. Then it calls the function `pascal()` to get the pascal triangle which it must print. Lastly, take care to deallocate the memory allocated for the pascal triangle.

V. Homework: TOKENS

Given a string, you must find the *words* it contains. The words are separated by white-spaces (i.e spaces, tabs, newlines). In more detail:

- traverse only once the initial string to find how many words it contains
- create (with `malloc()` an array with as many pointers needed that point to the beginning of each word
- each empty space character from the initial string must be replaced with the null characeter `\0`
- and finally, the array must have one more pointer with the value `NULL`, that indicates the end of the words

An example is illustrated in the following figure:



You must implement a function

```
char **string2words(char *str);
```

that performs the steps described above and returns the array with the pointers it created.

You must read the initial function in the `main()` function from the user using the `fgets()` function. Lastly, in a separate function (that is, not in main) print the words, one at a line.