Lab 8.8.6.2 Preprocessing: part 2

Objectives

Familiarize the student with:

- The preprocessor
- Printing on screen
- Functions
- Function calls with parameters
- Calling functions with parameters
- Function return values
- Printing on screen

Scenario

Write a function that checks whether or not a given string is a valid IP address (in human-readable form, of course). This function should return 1 if the address is valid, and 0 if not. Your function should check if:

- there are 4 parts in the string, separated by dots;
- each part contains only digits;
- each number is in the range of 0 to 255, inclusive.

For converting string fragments to integer values, you can use the *strtol*, *atoi* or *sscanf* functions. Separate the declaration of the function from its full definition. Write a second function that calls the first one and then prints an appropriate message: "127.0.0.1 is a valid IP address" or "a.b.c.d is not a valid IP address". Write a code to test it with values - call the second function. Do not limit yourself only to those values that we have given to you.

We did a similar task in Lab 6.6.11.2. The only difference between this lab and that one is that here you need to define all the constants (even one-character constants!) with preprocessor defines. You should also give them appriopriate names. After the whole process, you can (because you are an advanced user now) judge which option is better: to change all the possible constants or to leave some of them in their natural form.

```
#include <stdio.h>

/* your code */

int main()
{
 /* your code */
 return 0;
}
/* your code */
```

Example output

```
127.0.0.1 is a valid IP address
127.0.01 is not a valid IP address
127.0..1 is not a valid IP address
127.zero.0.1 is not a valid IP address
127.297.0.1 is not a valid IP address
127.2555.0.1 is not a valid IP address
```