French Azerbaijani University
Arash Habibi
Second year
2017-2018

.

Advanced C Programming

Practical Work 3 (a) : pointers and dynamic allocation

Sections 1, 2 consist of implementing and testing the questions of exercise session 4.

1 Passing arguments by reference

1. Write a function with prototype `int divide(float a, float b, float *q);` which divides `a` by `b` and puts the result at the `q` address. The returned value is a diagnosis (it says whether everything went right or not). The function returns 0 if `b=0` (impossible division) and 1 otherwise.
2. Write an example of how we can use this function to divide 365 by 5 and to print the result.

2 Dynamic allocation

1. Dynamically allocate an array of 5 integers ;
2. Dynamically allocate an array of 7 chars ;

3 Stack

In the last practical work session, you dealt with the following structure :

```
struct stack
{    int _size;
     int _array[300];
};
```

300 is the maximum number of elements that it is possible to store in the stack. We wish to modify this structure in order to be able to choose the maximum size of the stack. In other words, we don't want the maximum size to be always 300. We will need to take smaller or bigger stacks depending on user input. Here is the new structure :

```
struct stack
{    int _size;
     int *_array;
};
```

We also change the prototype of this function : `struct stack ST_new(int maxsize);`. Rewrite the `ST_new` function so that the `_array` field of the stack has `maxsize` elements.