Laboratory 11 – Pointers and arrays – part2

15.12.2020

1. [3points] Write a C function that swaps the values of two `float` variables. The type of swap function is void. The swap function has two parameters, pointers to `float`.

a) In the main function, declare two variables of the `float` t type. Initialize the variables using the `rand` function. Divide the value returned by `rand` by `RAND_MAX`. What to do to change the division type from `integer` to `float`?

b) Print the values of `float` variables to the screen.

c) Call the swap function and pass it the addresses of variables of type `float`.

d) Print the values of float variables to the screen.

```
Test data:
Before: x = 0.840188, y = 0.394383
After:  x = 0.394383, y = 0.840188
```

2. [7points] Swap and arrays.

a)Here is the main function. Fill in the blanks.

```
#include  <stdio.h>
#include  <stdlib.h>
#define SIZE 10

int main (void){
    float x[SIZE], y[SIZE];

    fill_array(…, …);
    fill_array(…, …);

    print_two_arrays(…, …, …);

    swap_arrays(…, …, …);

    printf("\n");
    print_two_arrays(…, …, …);

    swap_hlaf_arrays(…, …);

    printf("\n");
    print_one_array(…, …);

    return 0;
}
```

a) Write a function `fill_array` that takes two parameters: the address of the first element in the array and the address of the first element outside the array. The body of the function is two lines.

The first line is a `while` loop. The second line is to assign each element of the array a value generated by the rand function. The elements of the array are accessed using the first parameter, which is a pointer.

b) The `print_two_arrays` function will print all the elements of both arrays to the screen. The body of the function is two lines. Use a `for` loop. The function has the following parameters, the address of the first element in the first array, the address of the first element in the second array, number of elements.

c) The function `swap_arrays` swaps the values between the arrays passed to it when it is called. The body of the function is two lines. Use a `for` loop. The function has the following parameters, the address of the first element in the first array, the address of the first element in the second array, number of elements. In the `for` loop, call the `swap` function from the previous exercise.

d) The function `swap_hlaf_arrays` reverses the array passed to it when it is called. The body of the function is two lines. Use a `while` loop. The function `swap_hlaf_arrays` takes two parameters: the address of the first element in the array and the address of the last element in the array. In the `while` loop, call the `swap` function from the previous exercise.

e) The `print_one_arrays` function prints all the elements of an array to the screen. The body of the function is two lines. Use a `for` loop. The function has the following parameters: the address of the first element in the array and the address of the first element outside the array.

```
Test data:
x[0] = 0.840188, y[0] = 0.477397
x[1] = 0.394383, y[1] = 0.628871
x[2] = 0.783099, y[2] = 0.364784
x[3] = 0.798440, y[3] = 0.513401
x[4] = 0.911647, y[4] = 0.952230
x[5] = 0.197551, y[5] = 0.916195
x[6] = 0.335223, y[6] = 0.635712
x[7] = 0.768230, y[7] = 0.717297
x[8] = 0.277775, y[8] = 0.141603
x[9] = 0.553970, y[9] = 0.606969

x[0] = 0.477397, y[0] = 0.840188
x[1] = 0.628871, y[1] = 0.394383
x[2] = 0.364784, y[2] = 0.783099
x[3] = 0.513401, y[3] = 0.798440
x[4] = 0.952230, y[4] = 0.911647
x[5] = 0.916195, y[5] = 0.197551
x[6] = 0.635712, y[6] = 0.335223
x[7] = 0.717297, y[7] = 0.768230
x[8] = 0.141603, y[8] = 0.277775
x[9] = 0.606969, y[9] = 0.553970

x[0] = 0.606969
x[1] = 0.141603
x[2] = 0.717297
```

```
x[3] = 0.635712
x[4] = 0.916195
x[5] = 0.952230
x[6] = 0.513401
x[7] = 0.364784
x[8] = 0.628871
x[9] = 0.477397
```

Next lab 12 – Dynamic memory allocation.