Name: Kormoua Khongmeng
Neptun Code: I3MLPQ

Basic of Programming 2

Laboratory report 2

Task 1.

- **Is the code correct?**
  Yes, the code is correct. It just has a warning of using **scanf** function by visual studio.

- **Why do we use a function pointer?**
  Because we are using **qsort** which will call compare function to compare 2 variables and return a value back to **qsort** several times. When calling a function with a function like this, the value passed must be a pointer to a function. That is why we use function pointer here.

- **How can a function pointer be passed as a parameter?**
  We can pass a function pointer as a parameter by its name for example: "compare" or "&compare" is okay to reference as well base on a result that I got from modified the given code.

- **Why do the parameters of *compare* function have *void** types?**
  Because **qsort** is supposed to sort generic type, it can sort int, double, string, struct, etc. So, the *compare* function should not have a predefined data type for parameters, *void** is be a pointer which has no type but can point to somewhere and the specific data type that we will compare will be casted later inside the function. That is why we have *void** type here.

- **Why the last parameter of *scanf* should be a pointer?**
  Because in C, only the value will be copied and passed to the function. But what happen inside the function will not affect to the value of that variable at all. So, in order to write some data to some variable for example using a function **scanf** we need to pass an address or a pointer of that variable to the function, doing so **scanf** can write some data to that variable.

Task 2. Modify the given program so that the array size is not fixed anymore. Ask a user for the actual array size before reading the array elements. Consider dynamic memory management (recall it from C).

- The solution source code regarding to this question is in the same folder.
- Because of there are more than one source codes since there are more than one tasks needed to be solved, I named the source code for this question like this: **solution2.cpp**

Task 3. Having the examples above, define the most important usage areas of pointers.

- Base on the given example we can see that pointer can be used to handle various data types very well. It can make a function or a code to be universal. Pointer allows us to refer to some space that we don't know in advance what it is but can be handled efficiently and effectively.

Task 4. Develop and implement a function that receives some string value as a parameter and reverses it. In the implementation you cannot use any special functions from external string libraries.

- The solution source code regarding to this question is in the same folder.
- Because of there are more than one source codes since there are more than one tasks needed to be solved, I named the source code for this question like this: **solution4.cpp**