Restriction on pointer types: `char*`, `int*`, `void*`

dt@VPCF11M1E /2019/c++11/exercises/08_pointers_arrays_and_references/02

```
$ make build
g++ -std=c++11 -Wall -Wextra -I./include
./src/pointer_restrictions_on_this_system.cpp ./solution/main.cc
-o ./make/run.o
./src/pointer_restrictions_on_this_system.cpp:  In function void PointerRestrictionsOnThisSys
./src/pointer_restrictions_on_this_system.cpp:8:5:  error:  static assertion
failed:  int* cannot be casted to char* on this system
static_assert((sizeof(char*) < sizeof(int*)), "int* cannot be casted
to char* on this system");
^
./src/pointer_restrictions_on_this_system.cpp:12:5:  error:  static
assertion failed:  void* cannot be casted to void* on this system
static_assert((sizeof(char*) < sizeof(void*)), "void* cannot be casted
to void* on this system");
^
./src/pointer_restrictions_on_this_system.cpp:16:5:  error:  static
assertion failed:  void* cannot be casted to int*
static_assert((sizeof(int*) < sizeof(void*)), "void* cannot be casted
to int*");
^
makefile:26:  recipe for target 'build' failed
make:  *** [build] Error 1
```

It's clear that pointers are homogeneous, that is of P is a pointer of type T then P can be assigned to pointer Q if and only if the type of Q is T.

This cannot be stated in every cases when assigning a `void*` to a pointer of type T.