1. **Von Neumann architecture**

   Von Neumann architecture is the base of modern computer which programs and data are stored in a separate storage unit called memories and are treated the same. This make computer much easier to reprogram. There are three basic components :

   1. The Central Processing Unit (CPU)
   2. The Main Memory Unit
   3. The Input/Output Device

2. **Stored Program concept**

   A stored-program computer is a computer that stores program instructions in electronically or optically accessible memory. When at before, systems that stored the program instructions with plugboards or similar mechanisms.

3. Preprocess with gcc -E first.c

```
(base) yulongwang@YulongdeMBP Lec1src % gcc -E first.c
# 1 "first.c"
# 1 "<built-in>" 1
# 1 "<built-in>" 3
# 368 "<built-in>" 3
# 1 "<command line>" 1
# 1 "<built-in>" 2
# 1 "first.c" 2

# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/stdio.h" 1 3 4
# 64 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/stdio.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_stdio.h" 1 3 4
# 68 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_stdio.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h" 1 3 4
# 649 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/_symbol_aliasing.h" 1 3 4
# 650 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h" 2 3 4
# 715 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/_posix_availability.h" 1 3 4
# 716 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h" 2 3 4
# 69 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_stdio.h" 2 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/Availability.h" 1 3 4
# 135 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/Availability.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/AvailabilityVersions.h" 1 3 4
# 136 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/Availability.h" 2 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/AvailabilityInternal.h" 1 3 4
# 137 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/Availability.h" 2 3 4
# 70 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_stdio.h" 2 3 4

# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_types.h" 1 3 4
# 27 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_types.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/_types.h" 1 3 4
# 33 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/_types.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/machine/_types.h" 1 3 4
# 32 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/machine/_types.h" 3 4
# 1 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/i386/_types.h" 1 3 4
# 37 "/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/i386/_types.h" 3 4
typedef signed char __int8_t;



typedef unsigned char __uint8_t;
typedef short __int16_t;
typedef unsigned short __uint16_t;
typedef int __int32_t;
typedef unsigned int __uint32_t;
typedef long long __int64_t;
typedef unsigned long long __uint64_t;

typedef long __darwin_intptr_t;
```

```
(base) yulongwang@YulongdeMBP Lec1src % gcc -c first.c
(base) yulongwang@YulongdeMBP Lec1src % gcc -o first first.o
(base) yulongwang@YulongdeMBP Lec1src %
```

4. printf("main: a = %d, b = %d, argc = %d\n", a, b, argc);
   It works like printf in java. Where the three %d was replaced by a,b,argc. And then the whole string is output to std.