南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Advanced Programming

Lab 4, CMake, Inputs, Data Storage

廖琪梅, 王大兴, 于仕琪, 王薇

topics

- 1. CMake
- 2. Inputs
  - Command-Line Arguments
  - Standard Input
- 3. Data storage
  - array, string, struct, union
- 4. Exercises

What is CMake?

CMake is an open-source, cross-platform family of tools designed to build, test and package software. **CMake** is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.

For more information https://cmake.org/

CMake needs **CMakeLists.txt** to run properly.

A CMakeLists.txt consists of **commands** , **comments** and **spaces**.

- The **commands** include command name, brackets and parameters , the parameters are separated by spaces. Commands are not case sensitive.
- **Comments** begins with '#'.

Steps for generating a makefile and compiling on Linux using CMake:

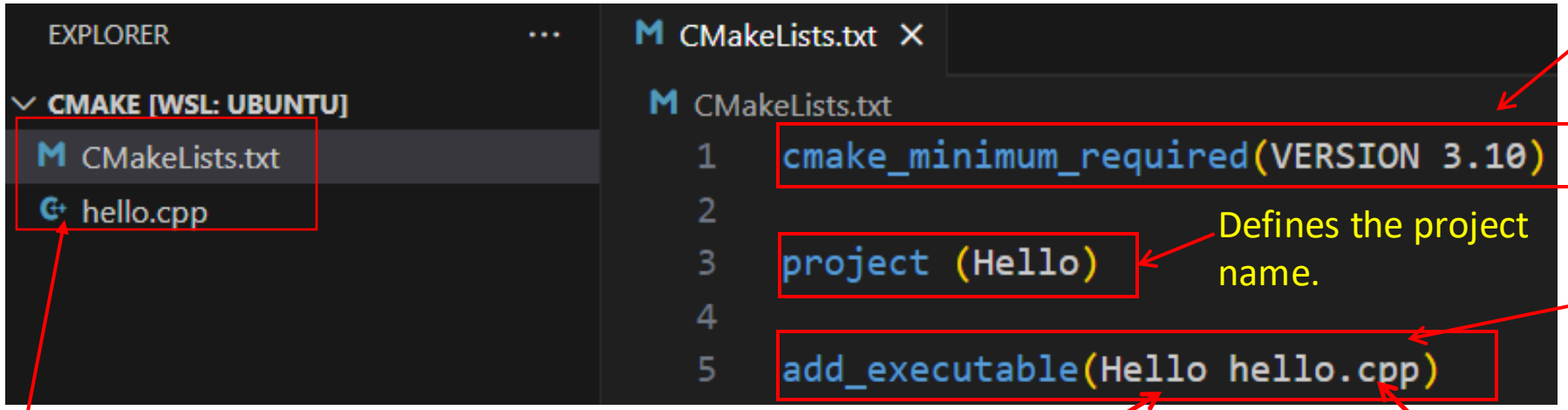
Step1: Writes the CMake configuration file **CMakeLists.txt**.

Step2: Executes the command **cmake PATH** to generate the **Makefile**. (PATH is the directory where the CMakeLists.txt resides.)

Step3: Compiles using the **make** command.

1. A single source file in a project

The most basic project is an executable built from source code files. For simple projects, a three-line **CMakeLists.txt** file is all that is required.



Specifies the minimum required version of CMake. Use **cmake --version** in Vscode terminal window to check the cmake version in your computer.

Defines the project name.

Adds the Hello executable target which will be built from hello.cpp.

The first parameter indicates the filename of executable file.

The second parameter indicates the source file.

Stores the CMakeLists.txt file in the same directory as the hello.cpp.

Suppose there is a hello.cpp

```
cmake > hello.cpp > ...
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Hello World!" << std::endl;
6  }
```

In current directory, type **cmake .** to generate makefile. If cmake does not be installed, follow the instruction to install cmake.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cmake .
```

```
Command 'cmake' not found, but can be installed with:

sudo apt install cmake
```

Install cmake first by instruction

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cmake .
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/CMake
```

Run cmake to generate makefle, . indicates the CMakeList.txt is in the current directory.

Makefile file is created automatically after running cmake in the current directory. Except Makefile, there are other new files and folders.

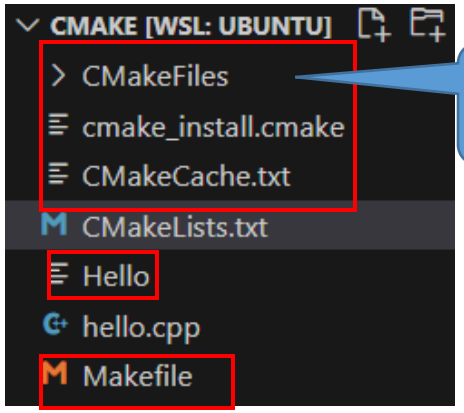
```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ ls
CMakeCache.txt  CMakeFiles  CMakeLists.txt  Makefile  cmake_install.cmake  hello.cpp
```

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ make
Scanning dependencies of target Hello
[ 50%] Building CXX object CMakeFiles/Hello.dir/hello.cpp.o
[100%] Linking CXX executable Hello
[100%] Built target Hello
```

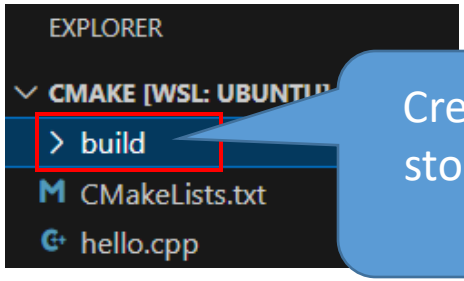
Execute make to compile the program.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ ./Hello
Hello World!
```

Run the program



Deletes all the building files and directory by CMake.



Creates an empty folder to store the building files and directory by CMake.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cd build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/CMake/build
```

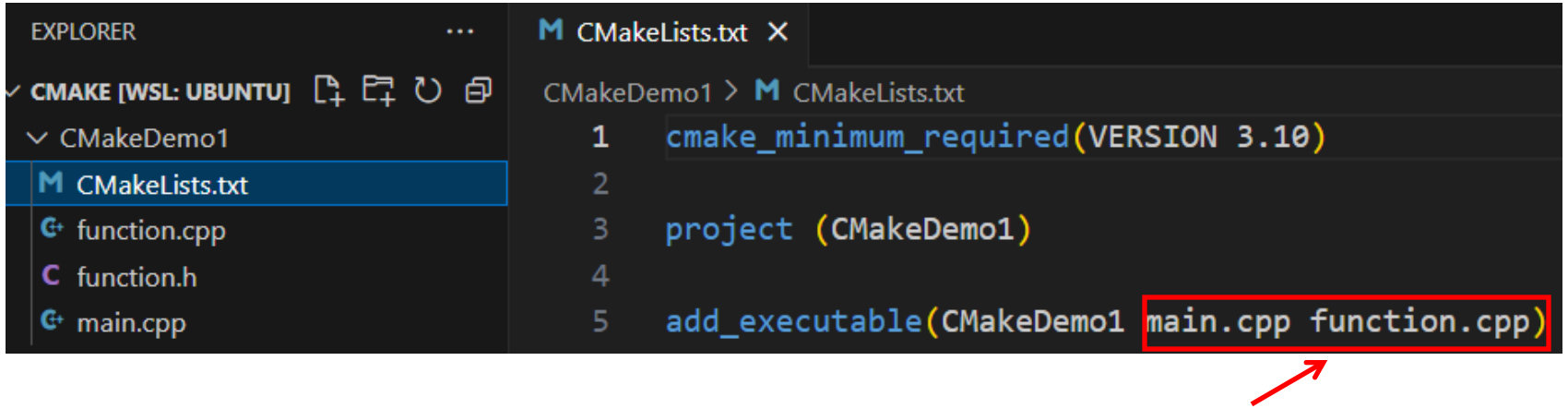
```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/build$ ls
CMakeCache.txt  CMakeFiles  Makefile  cmake_install.cmake
```

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/build$ make
Scanning dependencies of target Hello
[ 50%] Building CXX object CMakeFiles/Hello.dir/hello.cpp.o
[100%] Linking CXX executable Hello
[100%] Built target Hello
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/build$ ls
CMakeCache.txt  CMakeFiles  Hello  Makefile  cmake_install.cmake
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/build$ ./Hello
Hello World!
```

2. Multi-source files in a project

There are three files in the same directory.

```
./CmakeDemo1
      |
      +--- main.cpp
      |
      +--- function.cpp
      |
      +--- function.h
```



List all the source files using space as the separator.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cd CMakeDemo1
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo1$ mkdir build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo1$ cd build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo1/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/CMake/CMakeDemo1/build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo1/build$ make
Scanning dependencies of target CMakeDemo1
[ 33%] Building CXX object CMakeFiles/CMakeDemo1.dir/main.cpp.o
[ 66%] Building CXX object CMakeFiles/CMakeDemo1.dir/function.cpp.o
[100%] Linking CXX executable CMakeDemo1
[100%] Built target CMakeDemo1
```

Creates a folder

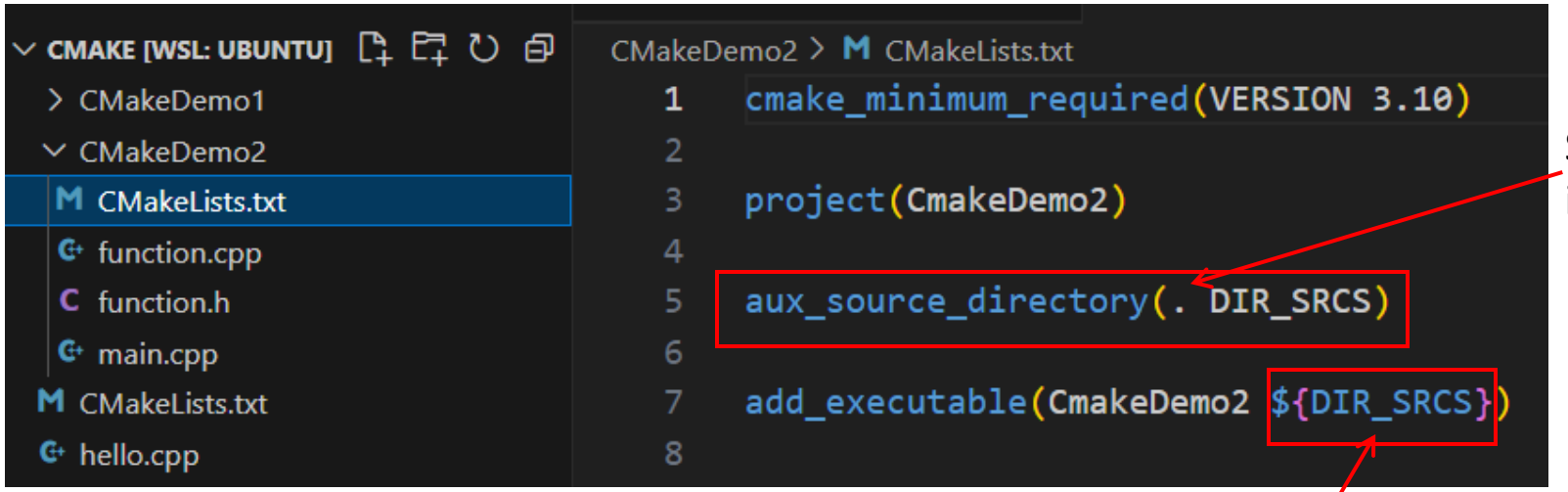
2. Multi-source files in a project-1

If there are several files in directory, put each file into the **add_executable** command is not recommended. The better way is using **aux_source_directory** command.

aux_source_directory (<dir> <variable>)

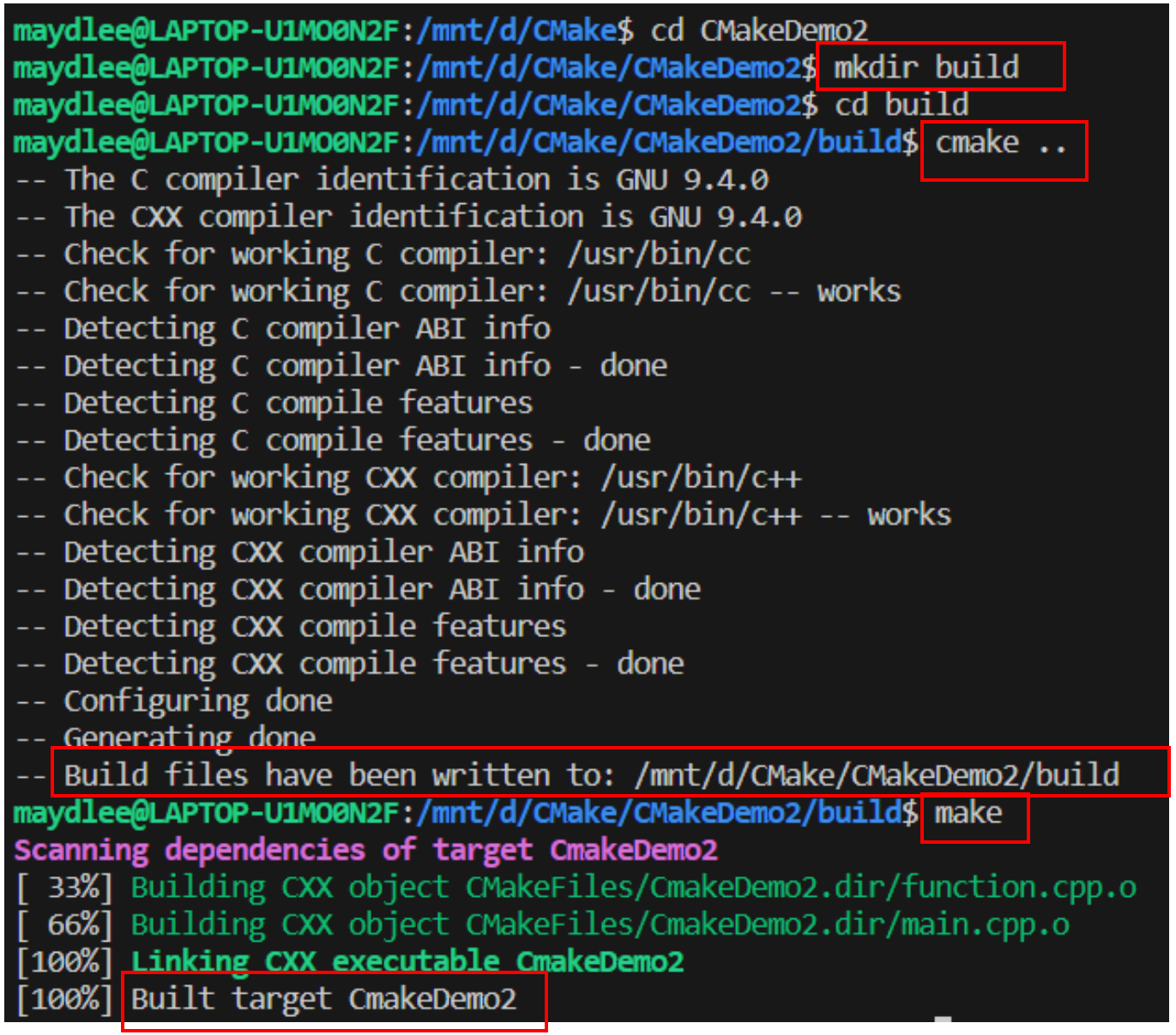
The command finds all the source files in the specified directory indicated by <dir> and stores the results in the specified variable indicated by <variable>.

2. Multi-source files in a project-2



Stores all files in the current directory into DIR_SRCS variable.

Compiles the source files in the variable by **${ }** into an executable file named CmakeDemo2

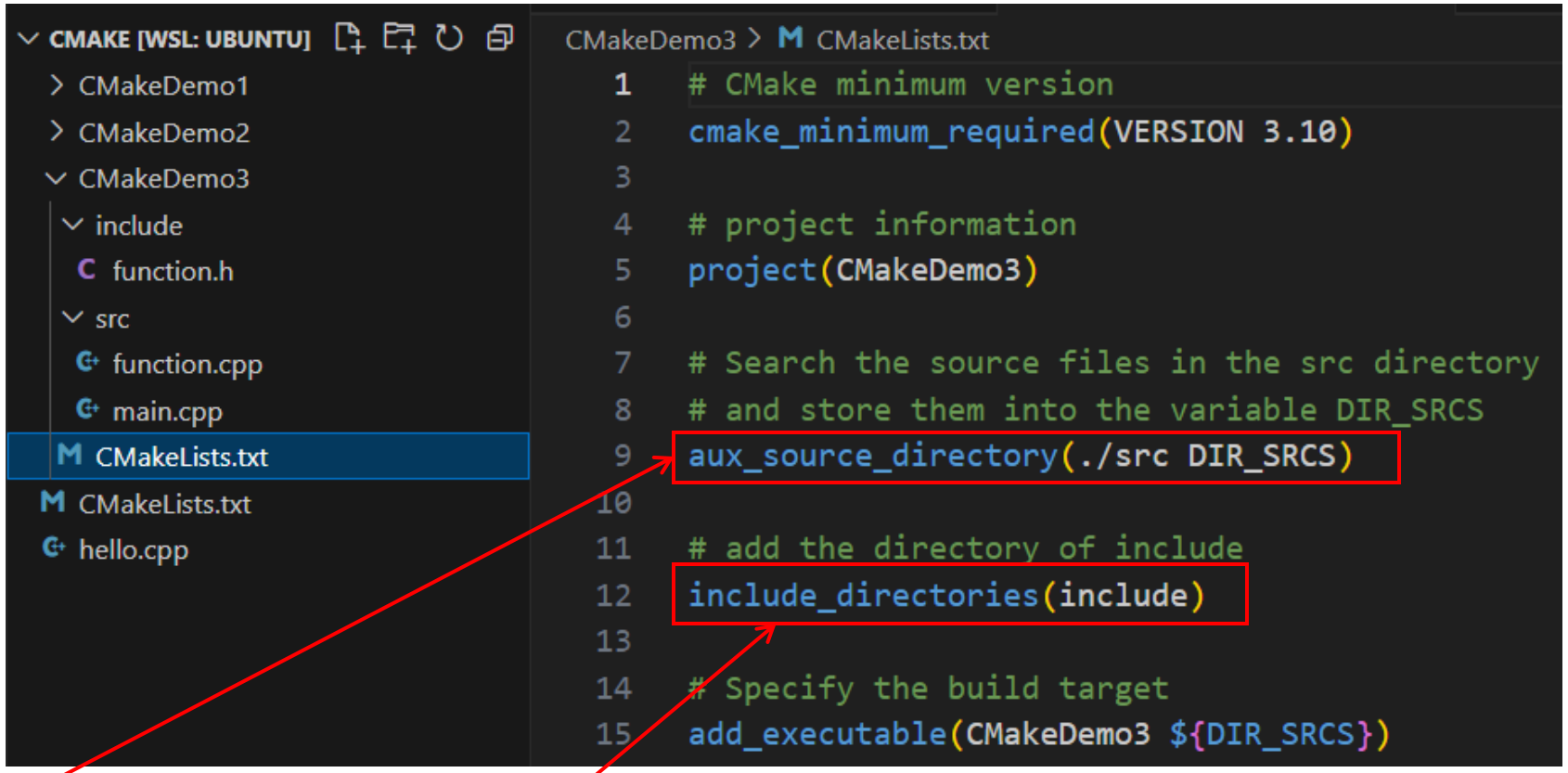

maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cd CMakeDemo2
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo2$ mkdir build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo2$ cd build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo2/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/CMake/CMakeDemo2/build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo2/build$ make
Scanning dependencies of target CmakeDemo2
[33%] Building CXX object CMakeFiles/CmakeDemo2.dir/function.cpp.o
[66%] Building CXX object CMakeFiles/CmakeDemo2.dir/main.cpp.o
[100%] Linking CXX executable CmakeDemo2
[100%] Built target CmakeDemo2

3. Multi-source files in a project in different directories

We write CMakeLists.txt in CmakeDemo3 folder.

./CMakeDemo3

```
|
+--- src/
|     |
|     +-- main.cpp
|     +-- function.cpp
|
+--- include/
      |
      +--- function.h
```



All .cpp files are in the **src** directory

Include the header file which is stored in **include** directory.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake$ cd CMakeDemo3
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo3$ mkdir build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo3$ cd build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo3/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/CMake/CMakeDemo3/build
maydlee@LAPTOP-U1MO0N2F:/mnt/d/CMake/CMakeDemo3/build$ make
Scanning dependencies of target CMakeDemo3
[ 33%] Building CXX object CMakeFiles/CMakeDemo3.dir/src/function.cpp.o
[ 66%] Building CXX object CMakeFiles/CMakeDemo3.dir/src/main.cpp.o
[100%] Linking CXX executable CMakeDemo3
[100%] Built target CMakeDemo3
```

For more cmake tutorial:

https://cmake.org/cmake/help/latest/guide/tutorial/index.html

https://riptutorial.com/cmake

2. Inputs

- 2. Inputs
  - 2.1 Command-Line Arguments
    - ✓ int main(int argc, char*argv[])
  - 2.2 Standard Input
    - ✓2.2.1 C style： scanf, gets vs fgets
    - ✓2.2.2 C++ style: cin, cin.gets vs cin.getline, getline()
  - others inputs
    - ✓ file, network, GUI, database, sensor

2.1 Command-Line Arguments

- At the beginning of program execution, arguments are read.
- All the arguments here are treated as string.
- Suitable for scenarios involving scripts and tools, but lacks interactivity.

```c
#include <stdio.h>   // c_a_demo.c

int main(int argc, char*argv[]){
   if(argc ==1)
      printf("ONLY argv[0]:%s\n",argv[0]);
   else
      for(int i=0;i<argc;i++)
         printf("argv[%d]: %s\n",i, argv[i]);

   return 0;
}
```

```
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ gcc c_a_demo.c
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ ./a.out
ONLY argv[0]:./a.out
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ ./a.out SUSTECH CSE CS219
argv[0]: ./a.out
argv[1]: SUSTECH
argv[2]: CSE
argv[3]: CS219
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo SUSTECH CSE CS219 | ./a.out
ONLY argv[0]:./a.out
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo SUSTECH CSE CS219 | xargs ./a.out
argv[0]: ./a.out
argv[1]: SUSTECH
argv[2]: CSE
argv[3]: CS219
```

2.2 Standard Input

- During program execution, read input data from standard input devices.
- Support different types of input data.
- Suitable for interacting with users.

```c
#include <stdio.h>

int main(int argc, char*argv[]){
   char uname[10]={""};
   char dname[10]={""};
   char cname[10]={""};
   printf("please input the name of University: ");
   scanf("%s", uname);
   printf("please input the name of department: ");
   scanf("%s", dname);
   printf("please input the name of course: ");
   scanf("%s", cname);
   printf("uname: %s, dname: %s, cname:%s\n",uname,dname,cname);
   return 0;
}
```

```
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ gcc c_in_demo.c
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ ./a.out
please input the name of University: SUSTECH
please input the name of department: CSE
please input the name of course: CS219
uname: SUSTECH, dname: CSE, cname:CS219
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo SUSTECH CSE CS219 | ./a.out
please input the name of University: please input the name of department: please input the name of course: uname: SUSTECH, dname: CSE, cname:CS219
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$
```

2.2.1 C style function about Standard Input processing

1-1. C: scanf

%d ----int

%f ----float

%c -----char

%s -----string

```c
#include <stdio.h>

int main()
{
    char str[20];
    printf("Enter a string:\n");
    scanf("%s", str);
    printf("You entered: %s\n",str);

    return 0;
}
```

There is no &

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/IO$ gcc scanf_printf.c
maydlee@LAPTOP-U1MO0N2F:/mnt/d/IO$ ./a.out
Enter a string:
Computer
You entered: Computer
maydlee@LAPTOP-U1MO0N2F:/mnt/d/IO$ ./a.out
Enter a string:
Computer Science
You entered: Computer
```

Why only Computer?

scanf uses whitespace—spaces, tabs, and newlines to delineate a string.

2.2.1 C style function about Standard Input processing

1-2. C: scanf

%d ----int

%f ----float

%c -----char

%s -----string

Tips:
When using scanf ("% d") or scanf ("% f") to read values, scanf skips leading whitespace characters (spaces, line breaks, etc.), but does not consume line breaks in the input stream (i.e., those generated by pressing enter).

```c
#include <stdio.h>

int main(){
   int prj_id=0;
   float prj_sc=0.0f;
   char valid=0;
   printf("please input 'project id' in decimal int: ");
   scanf("%d", &prj_id);
   printf("please input the score : ");
   scanf("%f", &prj_sc);
   printf("please input the score is valid or not(Y/N): ");
   while (getchar() != '\n');
   scanf("%c", &valid);
   printf("project id: %d, score: %.1f, %s\n",
          prj_id, prj_sc, (valid=='y'||valid=='Y')?"VALID":"NOT VALIDE" );
   return 0;
}
```

There is &

```
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ gcc c_in_dfc_format_demo.c
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ ./a.out
please input 'project id' in decimal int: 1
please input the score : 90.56
please input the score is valid or not(Y/N): y
project id: 1, score: 90.6, VALID
```

2.2.1 C style function about Standard Input processing

2. C: gets

```
fgets(str, 20, stdin);
```

```c
#include <stdio.h>

int main()
{
    char str[20];
    printf("Enter a string:\n");
    gets(str);

    printf("You entered: ");
    puts(str);

    return 0;
}
```

There is a warning due to using gets().
You can use fgets() function instead.

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/IO$ gcc gets_puts.c
gets_puts.c: In function ‘main’:
gets_puts.c:7:2: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
    7 |  gets(str);
      |  ^~~~
      |  fgets
/usr/bin/ld: /tmp/cc3CJcP4.o: in function `main':
gets_puts.c:(.text+0x34): warning: the `gets' function is dangerous and should not be used.
maydlee@LAPTOP-U1MO0N2F:/mnt/d/IO$ ./a.out
Enter a string:
Computer Science
You entered: Computer Science
```

Use gets to gain a sentence with a space. gets() stops reading input when it encounters a newline or end of file.

2.2.2 C++ style method about Standard Input processing

1. C++: cin

The cin is to use whitespace-- spaces, tabs, and newlines to separate a string.

```
Enter a string:C++
You entered: C++
Enter an other string:programming is funny.
You entered: programming
```

```
Enter a string:C++ programming is funny.
You entered: C++
Enter an other string:You entered: programming
```

cin_cout.cpp > ...

```cpp
#include <iostream>
using namespace std;

int main()
{
    char str[100];

    cout << "Enter a string:";
    cin >> str;
    cout << "You entered: " << str << endl;

    cout << "Enter an other string:";
    cin >> str;
    cout << "You entered: " << str << endl;

    return 0;
}
```

2.2.2 C++ style method about Standard Input processing

2. C++: cin.get( )

Input a single character:
istream& get(char&);
int get(void);

Input a string:
istream& get(char*,int);

cin_get.cpp > ...

```cpp
#include <iostream>
using namespace std;

int main()
{
    char str[20];

    cout << "Enter a string:";
    cin.get(str, 20);
    cout << "You entered: " << str << endl;

    cin.get();
    cout << "Enter an other string:";
    cin.get(str, 20);
    cout << "You entered: " << str << endl;

    return 0;
}
```

If the statement is omitted, what will be the output?

```
Enter a string:C and C++
You entered: C and C++
Enter an other string:C/C++ programming is funny.
You entered: C/C++ programming i
```

If the length of input string is greater than 20, it can only store first 19 characters in it.

```
Enter a string:C and C++
You entered: C and C++
Enter an other string:You entered:
```

2.2.2 C++ style method about Standard Input processing

3. C++: cin.getline( )

Input a string:
istream& getline(char*,int);

cin_getline.cpp > ...

```cpp
#include <iostream>
using namespace std;

int main()
{
    char str[20];

    cout << "Enter a string:";
    cin.getline(str, 20);
    cout << "You entered: " << str << endl;

    cout << "Enter an other string:";
    cin.getline(str, 20);
    cout << "You entered: " << str << endl;

    return 0;
}
```

```
Enter a string:C and C++
You entered: C and C++
Enter an other string:C/C++ programming is funny.
You entered: C/C++ programming i
```

If the length of input string is greater than 20, it can only store first 19 characters in it.

2.2.2 C++ style method about Standard Input processing

4. C++:

cin.get( ) vs cin.getline( )

getline() and get() both read an entire input line—that is, up until a newline character.

However, getline() discard the newline character, whereas get() leave it in the input queue.

```cpp
#include <iostream>
using namespace std;

int main()
{
    char str[20];

    cout << "Enter a string:";
    cin.get(str, 20);
    cout << "You entered: " << str << endl;

    cout << "Enter an other string:";
    cin.getline(str, 20);
    cout << "You entered: " << str << endl;

    return 0;
}
```

Program runs without entering another string

```
Enter a string:C and C++
You entered: C and C++
Enter an other string:You entered:
```

2.2.2 C++ style method about Standard Input processing

5. C++: string class I/O

getline() function takes the input stream as the first parameter which is cin and str as the location of the line to be stored.

```
Enter a string:C and C++
You entered: C and C++
Enter another string:C/C++ programming is funny.
You entered: C/C++ programming is funny.
```

string_input.cpp > ...

```cpp
#include <iostream>
using namespace std;

int main()
{
    string str;
    cout << "Enter a string:";
    getline(cin, str);
    cout << "You entered: " << str << endl;

    cout << "Enter another string:";
    getline(cin,str);
    cout << "You entered: " << str << endl;

    return 0;
}
```

2. Data storage

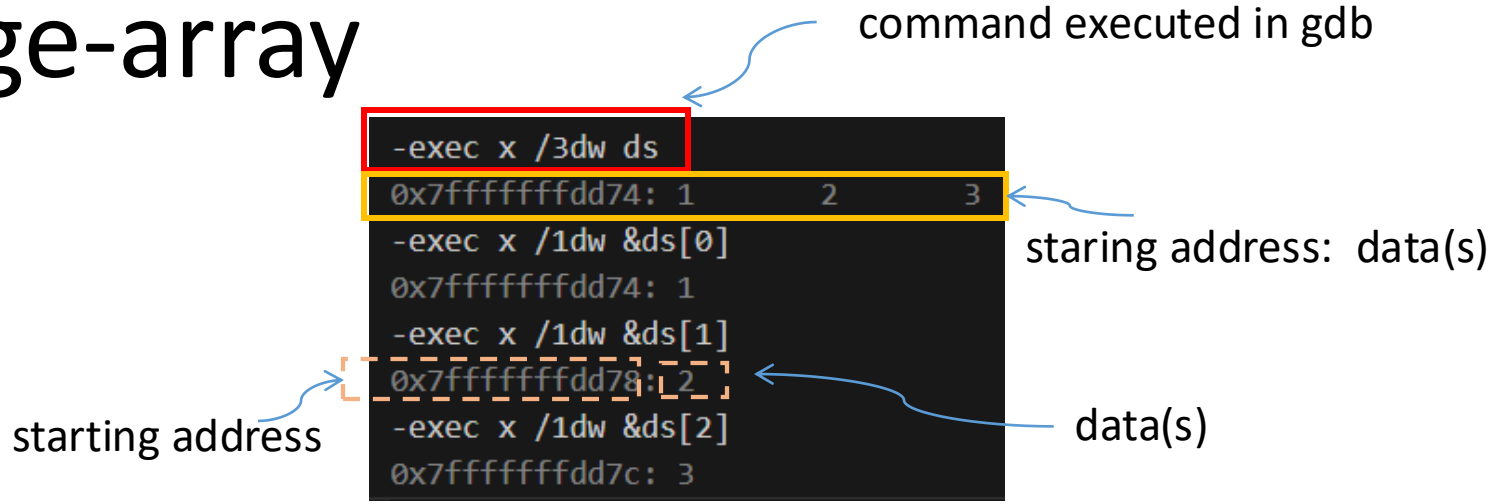
- 2.1.Data storage on construction type
  - array:
    - ✓One dimensional array and two-dimensional array
  - string:
    - ✓char array vs string
  - struct:
    - ✓align
  - union:
    - ✓share

2.1 Data storage-array

```
int main(){
    int ds[ ]={1,2,3 };

    int ds2 [2] [3]= {
        {4,5,6}, {7,8,9}
    };
    return 0;
}
```

```
ds: [3]
    [0]: 1
    [1]: 2
    [2]: 3
```




```
ds2: [2]
  [0]
    [0]: 4
    [1]: 5
    [2]: 6
  [1]
    [0]: 7
    [1]: 8
    [2]: 9
```

- using ‘x’ command in gdb to examine the data storage details
  - 1)address: starting from the position specified by the subsequent parameters
  - 2) datas: for example, option “**/3dw**” here means show the data stored in the space of **3** consecutive **w**ords starting from the address in **d**ecimal.

```
-exec x /6dw ds2
0x7fffffffdd80: 4       5       6       7
0x7fffffffdd90: 8       9
-exec x /3dw ds2[0]
0x7fffffffdd80: 4       5       6
-exec x /3dw ds2[1]
0x7fffffffdd8c: 7       8       9
```

```
-exec x /1dw &ds2[0][0]
0x7fffffffdd80: 4
-exec x /1dw &ds2[0][1]
0x7fffffffdd84: 5
-exec x /1dw &ds2[0][2]
0x7fffffffdd88: 6
-exec x /1dw &ds2[1][0]
0x7fffffffdd8c: 7
-exec x /1dw &ds2[1][1]
0x7fffffffdd90: 8
-exec x /1dw &ds2[1][2]
0x7fffffffdd94: 9
```

2.1 Data storage: char-array vs string

```
int main(){
    char SC[ ]="SUSTECH";
    char sc[ ]= {'s','u','s','t','e','c','h'};

    printf("size of SC: %ld bytes, sc: %ld\n",sizeof(SC), sizeof(sc));

    return 0;
}
```

```
size of SC: 8 bytes, sc: 7
```

```
-exec x /8cb SC
0x7fffffffdda0: 83 'S'  85 'U'  83 'S'  84 'T'  69 'E'  67 'C'  72 'H'  0 '\000'
-exec x /7cb sc
0x7fffffffdd99: 115 's' 117 'u' 115 's' 116 't' 101 'e' 99 'c'  104 'h'
```

The string terminator character(\000, value 0) is automatically included at the end of the string, but there is no such automatic operation in character arrays

2.1 Data storage: struct

```
#include<stdio.h>
struct data{
   int a;
   char c;
};

int main(){
   struct data icx;
   icx.a = 0x11223344;
   icx.c = 0x56;

   printf("size of icx: %ld\n",sizeof(icx));
   printf("size of icx.a: %ld, icx.a=0x%x\n",sizeof(icx.a),icx.a);
   printf("size of icx.c: %ld, icx.c=0x%x\n",sizeof(icx.c),icx.c);
   return 0;
}
```

```
size of icx: 8
size of icx.a: 4, icx.a=0x11223344
size of icx.c: 1, icx.c=0x56
```

```
-exec x /8xb &icx
0x7fffffffddc8: 0x44    0x33    0x22    0x11    0x56    0x00    0x00    0x00
-exec x /4xb &icx.a
0x7fffffffddc8: 0x44    0x33    0x22    0x11
-exec x /4xb &icx.c
0x7fffffffddcc: 0x56    0x00    0x00    0x00
```

Each member in the struct occupies exclusive space and is filled with necessary padding to achieve alignment.

2.2 Data storage: union

```c
#include<stdio.h>
union data
{
   int a;
   char c;
};

int main()
{
   union data endian;
   endian.a = 0x11223344;
   endian.c = 0x56;

   printf("size of endian: %ld\n",sizeof(endian));
   printf("size of endian.a: %ld,endian.a=0x%x\n",sizeof(endian.a),endian.a);
   printf("size of endian.c: %ld,endian.c=0x%x\n",sizeof(endian.c),endian.c);

    return 0;
}
```

```
size of endian: 4
size of endian.a: 4,endian.a=0x11223356
size of endian.c: 1,endian.c=0x56
```

```
-exec x /4xb &endian
0x7fffffffddcc: 0x56    0x33    0x22    0x11
-exec x /4xb &endian.a
0x7fffffffddcc: 0x56    0x33    0x22    0x11
-exec x /1xb &endian.c
0x7fffffffddcc: 0x56
```

All members in the union share the same space.

Exercise 1

Please refer to the content of courseware p14-p15 to generate a makefile using cmake tool and CMakeLists.txt, run the makefile to generate an executable file, and then run the executable file.

NOTES:
all the source files are in ./src , all the head files are in ./inc, all the build files are in ./build.

Exercise 2

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
   int cards[4]{};
   int hands[4];
   int price[] = {2.8,3.7,5,9,'C', "D"};
   char direction[4] {'L',82,'U',68};
   char title[] = "DeepSeek is an awesome tool.";
   cout << "sizeof(cards) = " << sizeof(cards) << ",sizeof of cards[0] = " << sizeof(cards[0]) << endl;
   cout << "sizeof(price) = " << sizeof(price) << ",sizeof of price[0] = " << sizeof(price[0]) << endl;
   cout << "sizeof(direction) = " << sizeof(direction) << ",length of direction = " << strlen(direction) << endl;
   cout << "sizeof(title) = " << sizeof(title) << ",length of title = " << strlen(title) << endl;

   return 0;
}
```

First, complete the code, then run the program, explain the result and answer the following question to SA. If it has bugs, fix them.

Q. It is asked to get the the number of characters in 'dirction'(which should be 4) by using strlen without changing the size of the 'dirction' array, one option is to add a piece of code between the definitions on "dirction" and "title":

A. char x = ' '; //a spce in ''

B. char x= 0;

C. char x='\0'

D. char xs[]=" " // a space in ""

E. other method

Exercise 3

```c
#include <stdio.h>  //p2.c

union data{
   int n;
   char ch;
   short m;
};

int main()
{
   union data a;
   printf("%d, %d\n", sizeof(a), sizeof(union data) );
   a.n = 0x40;
   printf("%X, %c, %hX\n", a.n, a.ch, a.m);
   a.ch = '9';
   printf("%X, %c, %hX\n", a.n, a.ch, a.m);
   a.m = 0x2059;
   printf("%X, %c, %hX\n", a.n, a.ch, a.m);
   a.n = 0x3E25AD54;
   printf("%X, %c, %hX\n", a.n, a.ch, a.m);

   return 0;
}
```

Run the program and explain the result to SA.

Exercise 4

- Complete the code on the right hand:
- 1) Design two enumeration types. The first is an enum “**Day**” for (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday), and the second is an enum “**Weather**” for (SUNNY, RAINY, CLOUDY, SNOWNY).
- 2) Complete the main function, which ask user to input the day value and the weather value accoding to the notice information, if the day is at weekend and the weather is SUNNY, pring out “can Travel”, else print out “not suitable for travelling”.
- The testing result is as shown on the next page.

```
#include <iostream>
using namespace std;
enum Day{/*complete code here if needed*/};
enum Weather{/*complete code here if needed*/};

int main( /*complete code here if needed*/ ){
   int d=0;
   int w=0;
   cout<<"input the Day value:
 Monday(1), Tuesday(2), Wednesday(3), Thursday(4),
Friday(5), Saturday(6), Sunday(7)\n";
  /*complete code here if needed*/

   cout<<"This is"<</*complete code here if needed*/<<endl;

   cout<<"input the Weather value: SUNNY(0), RAINY(1), CLOUDY(2),
SNOWNY(3)\n";
    /*complete code here if needed*/

   cout<<"The weather is: "<< /*complete code here if needed*/;

  if(/*complete code here if needed*/) cout<<"can Travel\n";
   else cout<<"not suitable for travelling\n";
   return 0;
}
```

Exercise 4

```
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ g++ p3.cpp
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo 6 0 | ./a.out
input the Day value: Monday(1), Tuesday(2), Wednesday(3), Thursday(4), Friday(5), Saturday(6), Sunday(7)
This is Saturday
input the Weather value: SUNNY(0), RAINY(1), CLOUDY(2), SNOWNY (3)
The weather is: SUNNY
can Travel
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo 7 0 | ./a.out
input the Day value: Monday(1), Tuesday(2), Wednesday(3), Thursday(4), Friday(5), Saturday(6), Sunday(7)
This is Sunday
input the Weather value: SUNNY(0), RAINY(1), CLOUDY(2), SNOWNY (3)
The weather is: SUNNY
can Travel
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo 1 0 | ./a.out
input the Day value: Monday(1), Tuesday(2), Wednesday(3), Thursday(4), Friday(5), Saturday(6), Sunday(7)
This is Monday
input the Weather value: SUNNY(0), RAINY(1), CLOUDY(2), SNOWNY (3)
The weather is: SUNNY
not suitable for travelling
ww2@DESKTOP-4NIH4UK:/mnt/c/Users/sustech/Desktop/C_CPP_CODE/lab4$ echo 6 7 | ./a.out
input the Day value: Monday(1), Tuesday(2), Wednesday(3), Thursday(4), Friday(5), Saturday(6), Sunday(7)
This is Saturday
input the Weather value: SUNNY(0), RAINY(1), CLOUDY(2), SNOWNY (3)
The weather is: unknow
not suitable for travelling
```