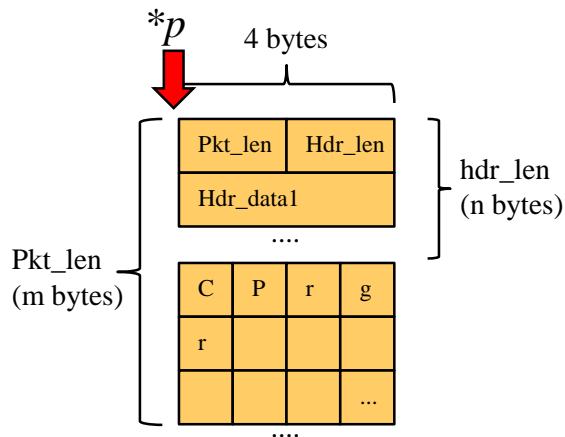


Week11, Task#9

- #9: 제공한 main 코드와 input.dat 파일을 사용하면, 아래 그림과 같이 패킷 정보를 불러들입니다.
- 포인터 p 와 포인터 연산만을 활용하여, 문자열로 저장된 패킷의 내용을 화면에 출력하는 프로그램을 작성하세요.
 - 포인터 p 는 패킷의 가장 첫부분을 포인팅.
 - 전체 패킷의 길이는 패킷의 0 byte offset 에 저장 (pkt_len)
 - 헤더의 길이는 패킷의 2 byte offset에 저장됨 (hdr_len)
 - 패킷의 내용은 hdr_len 과 pkt_len offset 사이에 저장됨.
- *Hello world*가 출력되어야 함.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define BUFLen 100

Void printpkt (char* p)
{
    short pkt_len = *(short*)p;
}
```

Your codes

```
int main(void)
{
    FILE *fp;
    char buf[BUFLen];
    char *p = buf;
    fp = fopen("input.dat", "rb");
    fread(buf, 1, BUFLen, fp);
    fclose(fp);

    printpkt (p);
}
```

Week11, Task#9

- #9: 아래 계산기가 구현된 main.c 파일이 올바르게 작동하도록 calc.c 와 calc.h 파일을 구현하세요.

main.c

```
#include <stdio.h>
#include "calc.h"

int main(void)
{
    int a = 2, b = 3;
    printf("A + B : %d\n", add (a, b));
    printf("A * B : %d\n", mul (a, b));
    printf("A^2 : %d\n", square (a));
}
```

calc.c

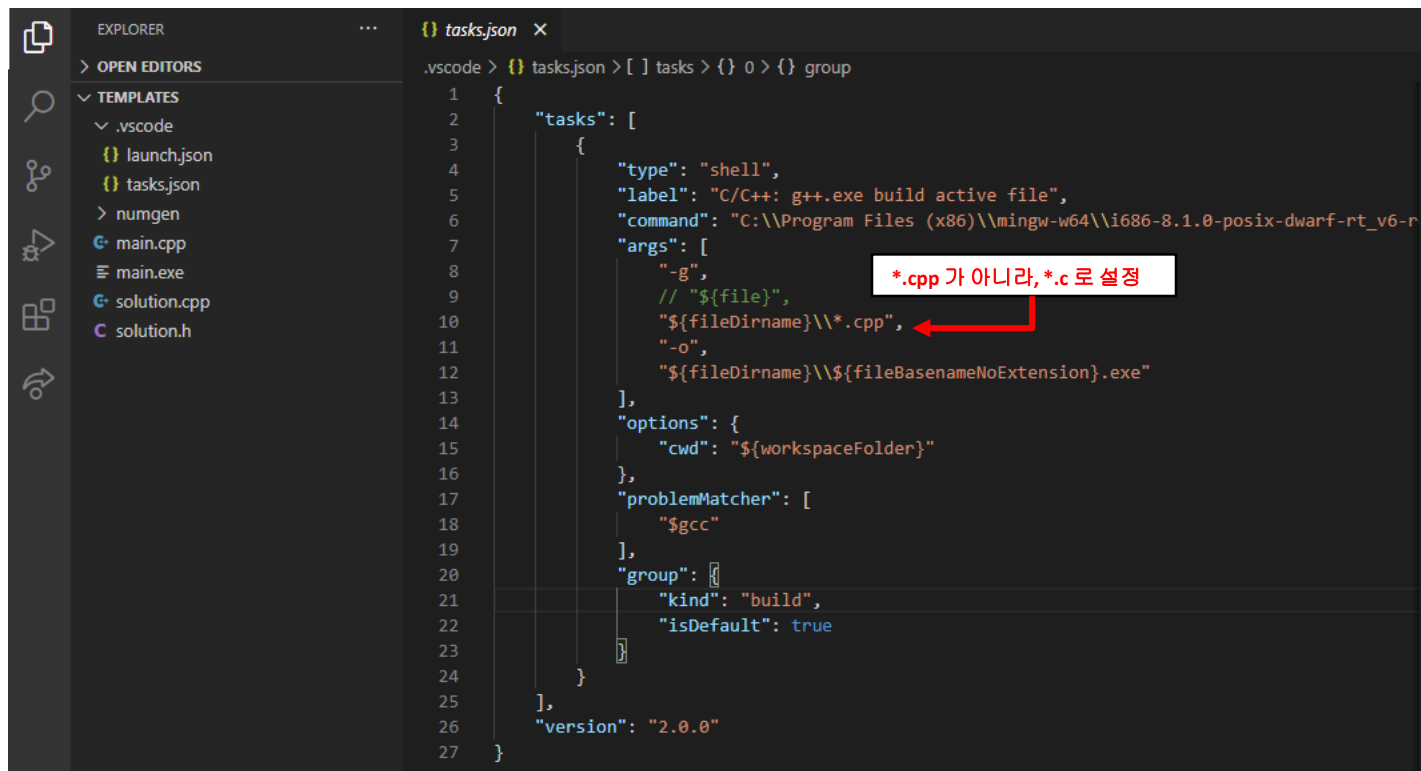
```
/* Fill up with your
codes */
```

calc.h

```
/* Fill up with your
codes */
```

Compiling multiple files on vscode

- Open the .vscode/tasks.json file
- Edit the “args” property.



```
.vscode > {} tasks.json > [] tasks > {} 0 > {} group
1 {
2   "tasks": [
3     {
4       "type": "shell",
5       "label": "C/C++: g++.exe build active file",
6       "command": "C:\\Program Files (x86)\\mingw-w64\\i686-8.1.0-posix-dwarf-rt_v6-r
7       "args": [
8         "-g",
9         // "${file}",
10        "${fileDirname}\\*.cpp",
11        "-o",
12        "${fileDirname}\\${fileBasenameNoExtension}.exe"
13      ],
14      "options": {
15        "cwd": "${workspaceFolder}"
16      },
17      "problemMatcher": [
18        "$gcc"
19      ],
20      "group": {
21        "kind": "build",
22        "isDefault": true
23      }
24    }
25  ],
26  "version": "2.0.0"
27 }
```

*.cpp 가 아니라, *.c 로 설정