ข้อมูลแบบแจกแจง
Enumeration Type

The enumeration type

- Set of named integer constants
- The named integer constants are called the “enumeration set”
- The values in an enumeration set don’t need to be sequential
- declared by the “enum” keyword
  - enum *tag_name* {*list of named integer constants*};
  - *tag_name* is a type specifier, such as

```
enum day { sunday, monday, tuesday, wednesday,
           thursday, friday, saturnday };
```


Declaration Examples

```
enum  temps {
                    zero =0,
                    freeze = 32,
                    boil = 220
};
enum  fruit    {    apple = 8,    pineapple,
                    pear = 3,     peach,        lemon
};
enum           { Green, White, Brown, Black };
```


Creating a variable of enumeration type

1. enum *tag_name* *variable list*;
   - enum temps water, gas;
   - enum fruit myfavor = peach;
2. enum {*list of named integer constants*} *variable list*;
   - enum { Green, White, Brown, Black } color;
3. enum tag_name {*list of named integer constants*} *variable list*;
   - enum day { sunday, monday, tuesday, wednesday, thursday, friday, saturnday } today, tomorrow;

```
/*possible values must come from type specifier */
today = friday;
tomorrow = saturnday;
if (today == tomorrow+1) …
```

Renaming type name with **typedef**

- typedef *ชื่อชนิดข้อมูลเดิม* *ชื่อชนิดข้อมูลใหม่*
  - typedef int Integer;
    Integer x;
  - typedef enum fruit Fruits;
    Fruits myfavor = peach;
  - typedef enum { Green, White, Brown, Black } Color;
    Color background, text;

Example I

```
/* Compute the next day*/
enum  day {sun, mon, tue, wed, thu, fri, sat};
typedef  enum day  Day;
Day FindNextDay(Day  d) {
        Day next_day;
        switch (d) {
                case sun     : next_day = mon;   break;
                case mon     : next_day = tue;   break;
                case tue     : next_day = wed;   break;
                case wed     : next_day = thu;   break;
                case thu     : next_day = fri;   break;
                case fri     : next_day = sat;   break;
                case sat     : next_day = sun;
        return (next_day);
}
```


Example II

```
/* Function to return number of days in a given month
 * Assume February has 28 days.
*/
enum month { jan=1, feb, mar, apr, may, jun, jul, aug,
                    sep, oct, nov, dec
};
typedef enum month Month;
int DayInMonth (Month  m) {
   switch (m) {
          case jan :
          case mar :
          case may :
          case jul :
          case aug :
          case oct :
          case dec :  return 31;
          case apr :  case jun :  case  sep : case nov : return 30;
          case feb :  return 28;
   }
}

printf("No. of day in this month is %d\n", DayInMonth(2));
```


แบบฝึกคิด (**ห้าม**ใช้ตัวแปรชุด)

- จงเขียนโปรแกรมเพื่อพิมพ์ ปฏิทินหนึ่งเดือน ตามที่ผู้ใช้ต้องการ โดยระบุเดือนและปีค.ศ. ที่ต้องการพิมพ์ปฏิทิน เช่น

```
Enter month number: 13
Month number be between 1 and 12, please re-enter: 2
Enter year: 11

February, 2011
S     M     T     W     Th    F     S
                  1     2     3     4     5
6     7     8     9     10    11    12
13    14    15    16    17    18    19
20    21    22    23    24    25    26
27    28
```