

# 計算機程式設計

## C語言 union enum

郭忠義

[jykuo@ntut.edu.tw](mailto:jykuo@ntut.edu.tw)

臺北科技大學資訊工程系

# 列舉集合 enum

- ❑ 列舉集合(enum)：列舉所有可能的元素形成一種資料型別
  - 限定變數的內容，使變數值可以名稱化。
  - 使程式可讀性高，利於撰寫智慧型程式。
  - 元素以整數儲存，預設從0開始，之後 +1，也可設定整數值

enum 型別名 {列舉所有元素} 變數名, 變數名, ...;

```
enum animal {tiger, horse, bird =100 , dog};
```

```
enum animal x; // x 是列舉型別變數，值只有四種可能
```

```
typedef enum animal {tiger, horse, bird =100 , dog} animal_t;
```

```
animal_t x; // x 是列舉型別變數，值只有四種可能
```

```
x = dog;
```

```
printf("%d", x); // 101
```

元素名	值
tiger	0
horse	1
bird	100
dog	101

# 列舉集合 enum

```
#include <stdio.h>
typedef enum animal {tiger, horse, bird , dog}
animal_t;
void test(animal_t x) {
    char *name[]={"tiger","horse","bird","dog"};
    printf("%s, ",name[(int)x]);
    if (x==tiger) printf ("Escape!\n");
    else if ( x==bird) printf("watch!\n");
    else printf("Go ....\n");
}
int main() {
    test(tiger);
    test(bird);
    test(horse);
    return 0;
}
```

```
tiger, Escape!
bird, watch!
horse, Go ....
```

# 列舉集合 enum

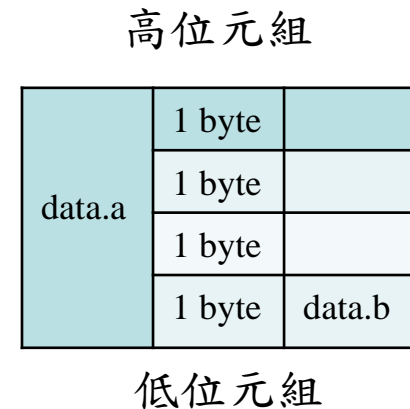
```
#include <stdio.h>
typedef enum grade_s{PASS = 60, GOOD = 80,
EXCELLENT = 90} grade_t;
void test(int score) {
    if (score>=EXCELLENT) printf("Wonderful~\n");
    else if (score>=GOOD) printf("Nice~\n");
    else if (score>=PASS) printf("Work Hard~\n");
    else printf("See you next year~\n");
}
int main() {
    test(59);
    test(90);
    test(85);
    test(60);
    return 0;
}
```

```
See you next year~
Wonderful~
Nice~
Work Hard~
```

# 共同空間 (union)

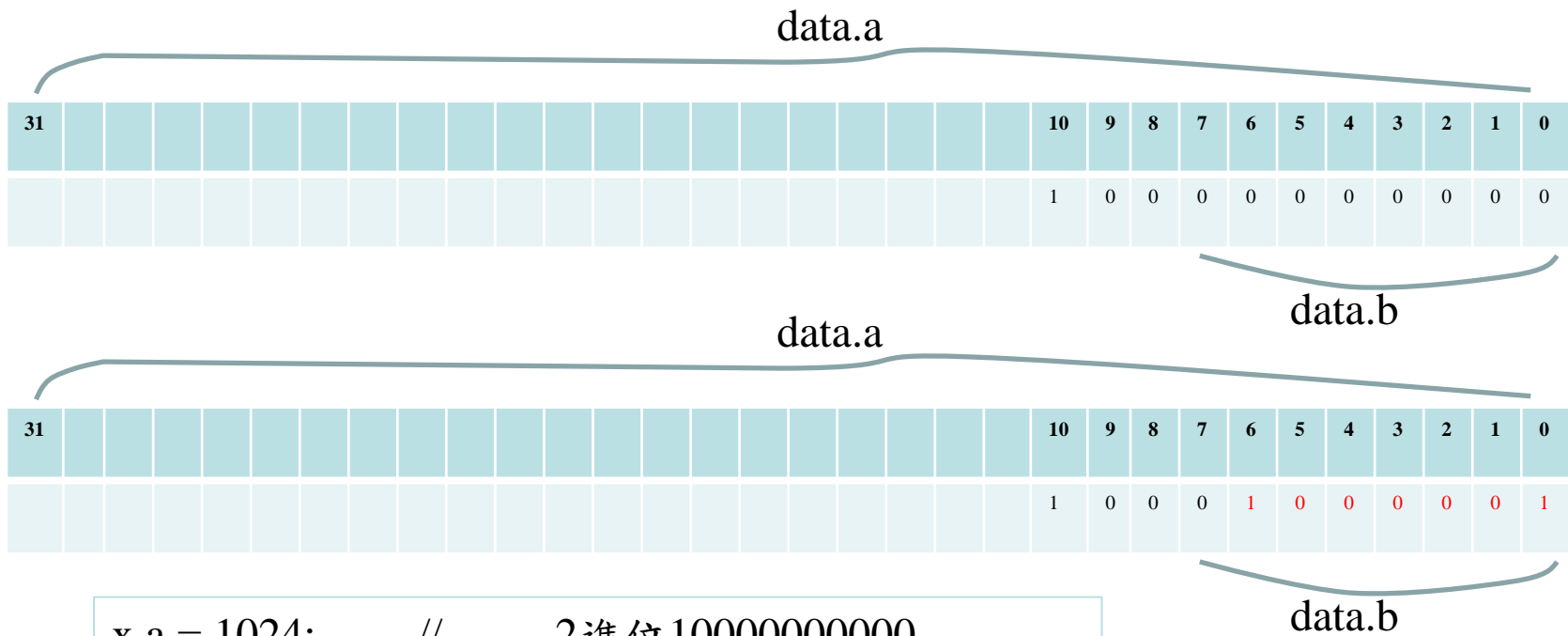
- ❑ 利用一段共用的空間來容納不同的資料型別。
  - 當一個變數的資料型別不確定時
  - 不同的資料型別存在相同的空間
  - 同時只能有一種資料型別存在

```
#include <stdio.h>
typedef union data {
    int a;  // 4 byte
    char b; // 1 byte
} data_t;  // 佔 4 byte
void test() {
    data_t x;
    x.a = 65;
    printf("%d\n", x.a); // 印出 65
    printf("%c\n", x.b); // 印出 A (編碼 65)
    x.a = 1024;
    x.b = 'A';
    printf("%d\n", x.a); // 印出 1089 = 1024 + 65
}
```



# 共同空間 (union)

- 利用一段共用的空間來容納不同的資料型別。
  - 兩個變數共用一個空間



```
x.a = 1024;    //      2進位100000000000
x.b = 'A';     // 65 的2進位10000001
printf("%d",x.a); // 最後變成 10001000001 = 1089
```

# 共同空間與列舉集合

□ 可選兩種不同評分方式 scoreType\_t {G, S}

○ grade\_t grade; (等第評分 G) {A, B, C}

○ int score; (百分評分 S) {0~100}

```
typedef enum scoreType_s {G, S} scoreType_t;
typedef enum grade_s {A, B, C} grade_t;
typedef union score_s {
    int score;
    grade_t grade;
} score_t; // 兩種評分選一種
```

// 印出評分結果

```
void print(score_t s, scoreType_t t) {
    char g[] = {'A', 'B', 'C'};
    if (t == G) printf("%c\n", g[s.grade]);
    else printf("%d\n", s.score);
}
```

```
void test() {
    int i=0;
    score_t student[3];
    scoreType_t type[3];
    char s[]={'A', 'B', 'C'};
    student[0].score=90;
    type[0]=S;
    student[1].grade=A;
    type[1]=G;
    student[2].grade=B;
    type[2]=G;
    for (i=0; i<3; i++) print(student[i], type[i]);
}
```

# Homework I

- ❑ 一個班級有N位同學(ID)，M門課，每一門課可以設定等地評分或百分制評分。可選兩種不同評分方式

- grade\_t grade;    (等第評分 G) {A, B, C}

- int score;        (百分評分 S) {0~100}

```
typedef enum scoreType_s {G, S} scoreType_t;
typedef enum grade_s {A, B, C} grade_t;
typedef union score_s {
    int score;
    grade_t grade;
} score_t;        // 兩種評分選一種
```

- 請使用enum定義不同評分制資料型別，使用union定義分數資料型別。
- 設定M門課評分制，輸入N位同學M門課成績，根據下表換算，計算每一位同學百分制平均成績，印出前三名ID與平均成績，四捨五入到整數。



# Homework I

## ○ 分數對照表

等第	GPA	百分制區間	百分制對照
A+	4.3	90-100	95
A	4.0	85-89	87
A-	3.7	80-84	82
B+	3.3	77-79	78
B	3.0	73-76	75
B-	2.7	70-72	70
C+	2.3	67-69	68
C	2.0	63-66	65
C-	1.7	60-62	60
F	0	59以下	50
X	0	0	0