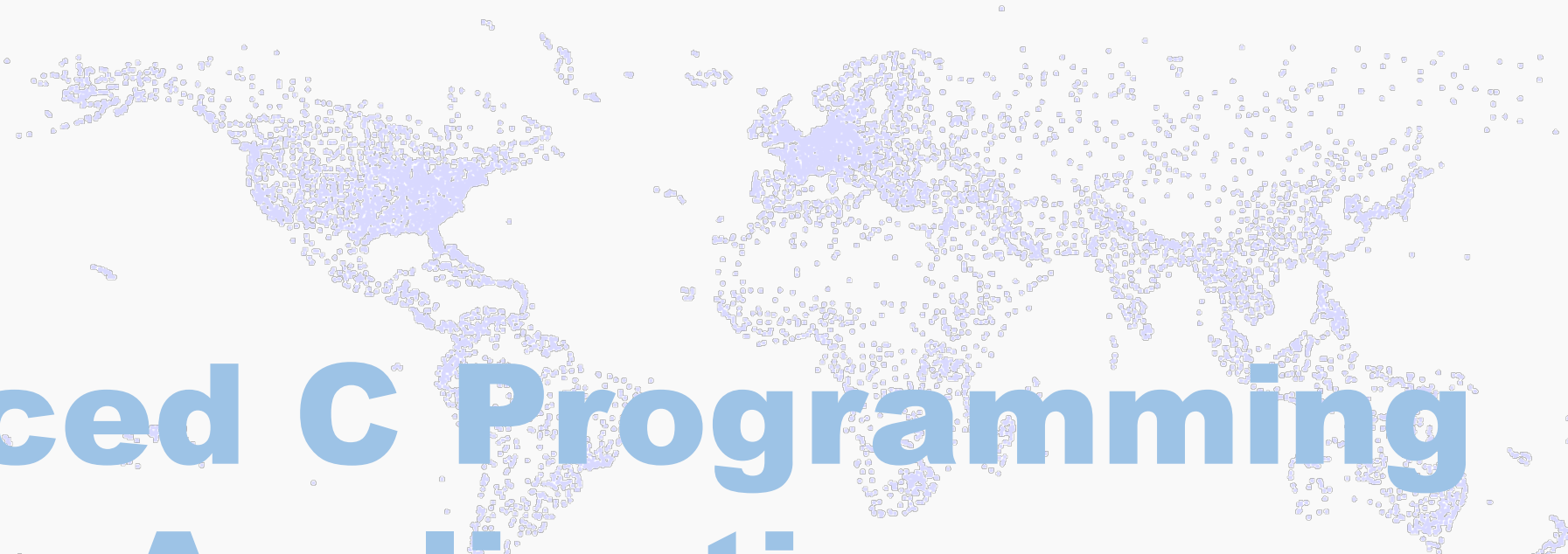


<Adv C & App/>



Advanced C Programming And It's Application

Review: Condition & Array

Assistant Prof. Chan, Chun-Hsiang

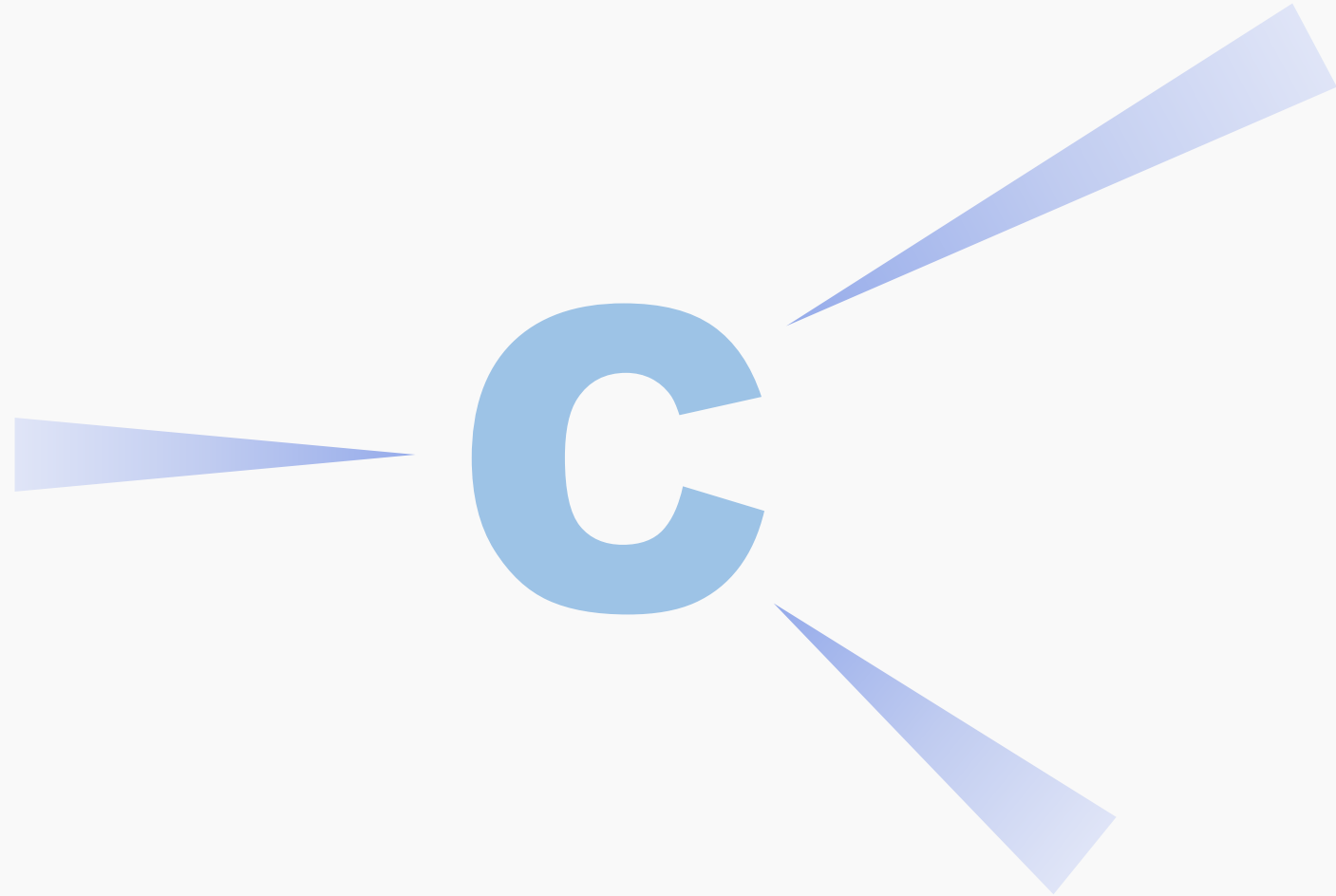
Department of Artificial Intelligence, Tamkang University

Oct. 6, 2021

</ Adv C & App >

大綱

- [1] Conditions
- [2] If else
- [3] Switch case
- [4] Control
- [5] 1-D Array
- [6] 2-D Array
- [7] Assignments



<Conditions/>

條件判斷式

條件判斷式的功能就是可以讓程式設計者，設定在某一條件下，程式會自動做某件事情，例如：

```
if district=="xinyi"  
    city="Taipei"  
else  
    city="NewTaipei"
```

<If else/>

If ... else ...

if else式其中一種條件判斷式，可以利用if, else if, else一直做判斷下去。

```
if (condition expression){  
    // statement  
} else if (condition expression){  
    // statement  
} else {  
    //statement  
}
```

<If else/>

If ... else ...

```
/*Ex 2-1: If else */
/* If else if else */
int age;

printf("Ex 2-1: If else\n");
printf("How old are you?\n");
scanf("%d", &age);
if (age<18){
    printf("He or she is not yet an adult!\n");
} else if (age<20){
    printf("He or she cannot vote yet!\n");
} else{
    printf("He or she can vote and bear full legal
responsibility!\n");
}
</If else>
```

<If else/>

If ... else ...

Lab 2-1:

用年齡判斷學歷:

07 < y.o. <= 12: 國小

12 < y.o. <=15: 國中

15 < y.o. <= 18: 高中

18 < y.o. < 22: 大學

/*Ex 2-2: Multiple If else */

/* Multiple If else if else */

int age;

printf("Ex 2-2: Multiple If else\n");

printf("How old are you?\n");

scanf("%d", &age);

if (age < 12){

printf("You are child!");

} else if (age >= 12 && age < 20){

printf("You are adolescent!");

} else if (20 <= age < 65) {  It does not work correctly!

printf("You are middle age!");

} else {

printf("You are old people!");

}

</If else>

Switch... case...

Switch case 與 if else 不一樣的地方是switch case是依照不同的類型執行不同的code。

```
switch (condition expression) {  
    case CASE1:  
        //statement  
        break;  
    default:  
        //statement  
        break;  
}
```

<Switch case/>

Switch... case...

Switch case 與 if else 不一樣的地方是switch case是依照不同的類型執行不同的code。

```
/*Ex 2-3: Switch case */
/* Switch case */
char gpa;

printf("Ex 2-2: Switch case\n");
printf("Wanna to know your score? Key in your
GPA here!\n");
scanf("%c", &gpa);
```

```
switch (gpa){
    case 'A':
        printf("Your score ranges from 90-100!");
        break;
    case 'B':
        printf("Your score ranges from 80-90!");
        break;
    case 'C':
        printf("Your score ranges from 70-80!");
        break;
    case 'D':
        printf("Your score ranges from 60-70!");
        break;
    default:
        printf("Your score belows 60!");
        break;
}
```


Switch... case... or if... else...

Lab 2-2:

遊樂園門票販售機:

成人門票: 100 NT

學生(< 18 y.o.)、老人(< 65 y.o.)、軍警: 50 NT

小孩 (< 10 y.o.): 0 NT

撰寫一個程式，讓使用者輸入其身分或年齡，判斷門票的價格為多少元。

break; continue; goto

break是可以直接中止迴圈並跳出

continue是可以直接跳過

goto可以在程式中預先設定好的標籤中作跳躍

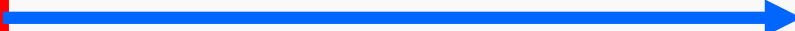
可能會稍微有點難想像，我們就來做三個實驗觀察這其中的差異。

Lab 2-3:

設計一個迴圈，使 `a` 可以從一跑到五，設定每次印出 `a` 的現況，當 `a = 3` 的時候先是放一個 `break` 或是 `continue` 並印出一個 `Hello world`，最後在外部寫一個 `ending`。

<Control/>

break; continue; goto

```
for (int a = 0; a < 5; a++) {  
    printf("%d\t", a);  
    if (a == 3) {  
        break;  continue;  
        printf("Hello world!\n");  
    }  
}  
printf("Ending!");
```

<Control/>

goto

/*Ex 2-6: Control - goto */

int count = 0;

printf("Ex 2-6: Control - goto\n");

START:

for (int a = 0; a < 5; a++) {

printf("%d\t", a);

if (a == 3 && count < 3) {

printf("Hello world!\n");

count += 1;

goto START;

} else {

continue;

}

}

printf("Ending!");

</Control/>

<1-D Array/>

一維矩陣

int intArr[10]

index	0	1	2	3	4	5	6	7	8	9
memory	1	2	3	4	5	6	7	8	9	10

在C語言的矩陣，可以定義程很多不同的資料格式。
資料型態 名稱[大小] | e.g., int intArr[10];

#define LENGTH 10

/*Ex 2-7: 1-D array */

/* 1-D array */

int arr[LENGTH] = {0};

int i, c = 0;

printf("Ex 2-3: 1-D array\n");

printf("original array value: ");

for(i = 0; i < LENGTH; i++) {

printf("%d ", arr[i]);

}

printf("\nsum of value from the first");

```
for(i = 0; i < LENGTH; i++) {
    c += i;
    arr[i] = c;
}
```

```
printf("\nshow result: ");
for(i = 0; i < LENGTH; i++) {
    printf("%d ", arr[i]);
}
putchar('\n');
```

Lab 2-4:

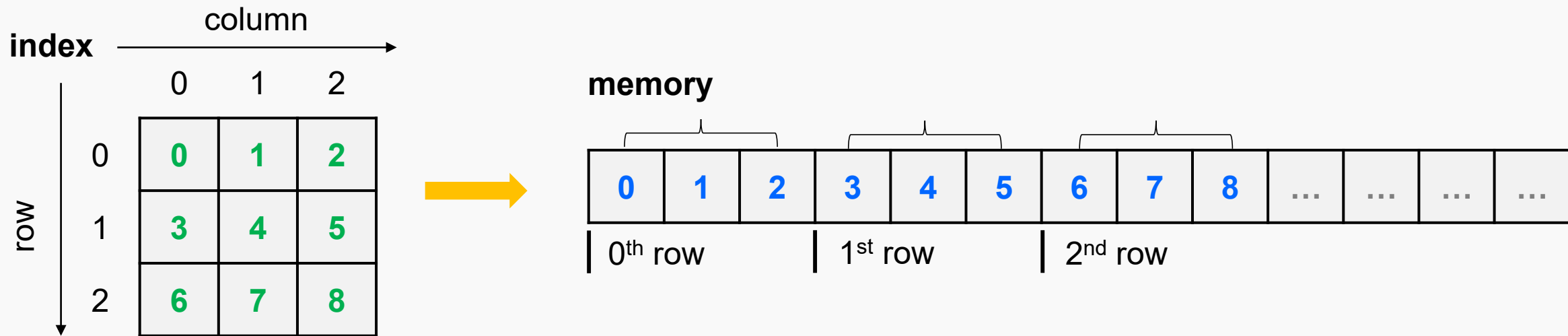
int num[10] = {0};

double dou[10] = {0.0};

char cha[10] = {'\0'};

<2-D Array/>

二維矩陣

圖解二維矩陣 matrix \rightarrow memory space

Row major concept

$$\text{arr}[0][0] = 0$$

$$\text{arr}[1][2] = 5$$

$$\text{arr}[2][1] = ?$$

二維矩陣

```
#define ROW 4  
#define COLUMN 6
```

```
/*Ex 2-8: 2-D array */
```

```
/* 2-D array */
```

```
int arr[ROW][COLUMN] = {0};
```

```
int i, j;
```

```
printf("Ex 2-4: 2-D array\n");
```

```
printf("original array value: ");
```

```
for(i = 0; i < ROW; i++) {
```

```
    for (j = 0; j < COLUMN; j++){
```

```
        arr[i][j] = i*j;
```

```
    }
```

```
}
```

```
int arr[4][6]
```

index	0	1	2	3	4	5	6	7	8	9
memory	1	2	3	4	5	6	7	8	9	10

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23

```
printf("show result: \n");
```

```
for(i = 0; i < ROW; i++) {
```

```
    for (j = 0; j < COLUMN; j++){  
        printf("%d\t", arr[i][j]);
```

```
    }
```

```
    putchar('\n');
```

```
}
```

<Assignments/>

作業一

製作一個程式，可以幫忙計算YouBike的租借價格。

參考資料:

https://www.dot.gov.taipei/News_Content.aspx?n=8AF5C9CCCF7B8610A&sms=87415A8B9CE81B16&s=7F22D0A704345945

公共運輸定期票之使用者，每次騎乘亦享前30分鐘費率免費之優惠。

騎乘逾30分鐘，但於4小時內還車，費率為每30分鐘10元。

騎乘逾4小時，但於8小時內還車，第4~8小時費率為每30分鐘20元。

騎乘逾8小時，於第8小時起將以每30分鐘40元計價。

單次租車的收費為

騎乘4小時內，費率為每30分鐘10元。

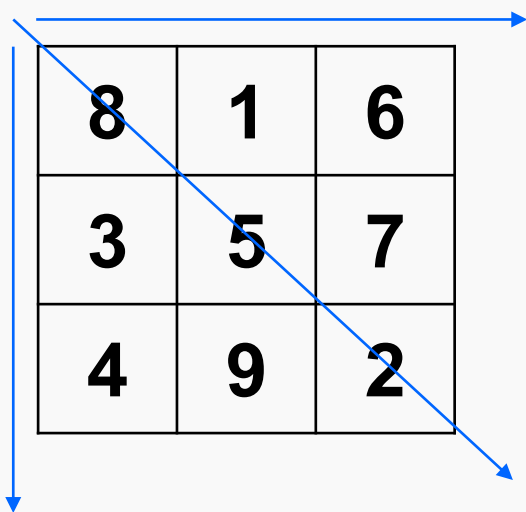
騎乘逾4小時，但於8小時內還車，第4~8小時費率為每30分鐘20元。

Notice: 預設為無公共運輸定期票

<Assignments/>

作業二

有聽過 **magic array**? 不論是按列加總、按行加總、甚至是對角線加總都會是相同的數值。



8	1	6
3	5	7
4	9	2

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

請定義一個矩陣，把n=4的**magic array**數值存進去，並再印出來。

References

<https://openhome.cc/Gossip/CGossip/index.html>

<https://edisonx.pixnet.net/blog/post/35305668>

<https://www.learn-c.org/>

<http://tw.gitbook.net/cprogramming/>