

資料結構之 C 語言重點複習（一）

1.基本資料型態與位址

- sizeof(int),sizeof(float),sizeof(char)大小
- sizeof(a),sizeof(b),sizeof(c)大小

```
-----  
int a;float b;char c;  
printf(“%d %d %d\n”, sizeof(int),sizeof(float),sizeof(char));  
printf(“%d %d %d\n”, sizeof(a),sizeof(b),sizeof(c));  
-----
```

- 宣告 int a 後，a 與&a 的差別？

```
-----  
int a=3;float b=5.834;char c='A';int d=99;  
printf(“%d %.2f %c %d\n”,a,b,c,d);  
printf(“%d %d %d %d\n",&a,&b,&c,&d);  
scanf(“%d %d”,a,d);  
scanf(“%d %d",&a,&d);  
-----
```

2.陣列

- 宣告 char str[100]後，str 與&str 的差別？

```
-----  
char str[100]=“abcd1234”;  
printf(“%s\n”,str);  
printf(“%s\n",&str);  
printf(“%d %d\n”,str,&str);  
scanf(“%s”,str);  
printf(“%s\n”,str);  
scanf(“%s",&str);  
printf(“%s\n”,str);  
-----
```

- 宣告 char str[100]後，str 中任一 element 的操作？

```
-----  
scanf(“%c",&str[2]);  
printf(“%s\n”,str);
```

```
for (i=0;i< strlen(str);i++)//正向印 str, 呼叫 strlen()必須 include <string.h>  
    printf(“%c”,str[i]);
```

```
printf("\n");
for (i=strlen(str)-1;i>=0;i--)//反向印 str
    printf("%c",str[i]);
printf("\n");
```

```
str[2]='\0';//或 str[2]=0
printf("%s\n",str);
```

- 宣告 int arra[4]後，如何操作？

```
int arra[4];

for (i=0;i<4;i++)
    printf("%d %d\n",&arra[i],arra[i]);

for (i=0;i<4;i++)
    scanf("%d",&arra[i]);

for (i=0;i<4;i++)
    printf("%d %d\n",&arra[i],arra[i]);
```

3.結構

- 先宣告二整數（num，score）在同一結構 ASTUDENT 中，再另外宣告結構 ASTUDENT 的變數 onestudent

```
struct ASTUDENT{
    int num,score;
};
struct ASTUDENT onestudent;
sizeof(struct ASTUDENT);// ? ? ? sizeof(ASTUDENT) ;
sizeof(onestudent);
```

- 使用結構 ASTUDENT 變數 onestudent 中的成員變數

```
printf("%d %d %d %d\n", &onestudent.num, onestudent.num,
```

```
&onestudent.score, onestudent.score);
scanf("%d %d",&onestudent.num, &onestudent.score);
printf("%d %d %d %d\n", &onestudent.num, onestudent.num,
&onestudent.score, onestudent.score);
-----
```

- 可宣告二整數（num，score）在結構 ASTUDENT 中，再一併宣告結構 ASTUDENT 的變數

```
-----
struct ASTUDENT{
    int num,score;
} onestudent;
-----
```

4.結構陣列

- 先宣告二整數（num，score）在同一結構 ASTUDENT 中，再另外宣告結構 ASTUDENT 的陣列變數 fourtudent

```
-----
struct ASTUDENT{
    int num,score;
};
struct ASTUDENT fourstudent[4];
```

```
sizeof(fourstudent);// ? ?
sizeof(fourstudent[2]); // ? ?
-----
```

- 使用結構 ASTUDENT 陣列變數 fourstudent 中的成員變數

```
-----
for (i=0;i<4;i++)
    printf("%d %d %d %d\n", &fourstudent[i].num, fourstudent[i].num,
&fourstudent[i].score, fourstudent[i].score);

for (i=0;i<4;i++)
    scanf("%d %d", &fourstudent[i].num, &fourstudent[i].score);
```

```
for (i=0;i<4;i++)
    printf(“%d %d %d %d\n”, &fourstudent[i].num, fourstudent[i].num,
    &fourstudent[i].score, fourstudent[i].score);
-----
```

- 結構 ASTUDENT 中，再一併宣告結構 ASTUDENT 的變數

```
-----
struct ASTUDENT{
    int num,score;
} fourstudent[4];
-----
```

5.邏輯實現

- 問題：輸入變數 n 值，印出 n 列，第一列有一個星號，每後一列多二個星號，正中對齊。
- 想法：必須使用雙層迴圈架構處理
- 推理：外層迴圈處理每一列，內層迴圈處理每一列所需之空白及星號
- 實做：

```
-----
for (i=0;i<n;i++){
    for (j=1;j<=n-1-i;j++) printf(“ “); //印出每列所需的空白格數
    for (j=0;j<(2*i+1);j++) printf(“*“); //印出每列所需的星號數
    printf(“\n”);
}
-----
```

6.函數

- 呼叫前先要先定義（一）

```
-----
int adding(int a,int b){
    return a+b;
}

main(){
    int i=3,j=5;
    printf(“%d+%d=%d\n”,i,j,adding(i,j));
}
-----
```

- 呼叫前先要先定義（二）

```
int adding(int,int); //定義列，或 int adding(int a,int b);
```

```
main(){
    int i=3,j=5;
    printf(“%d+%d=%d\n”,i,j,adding(i,j));
}
```

```
int adding(int a,int b){
    return a+b;
}
```

7.遞迴函數

- 問題：以遞迴方式計算 $1+2+\dots+n$
- 想法：寫成 $f(n)=f(n-1)+n$ ， $f(1)=1$
- 推理：
 - ◆ 最基礎定義遞 $f(1)=1$ 成為遞迴函數的停止條件
 - ◆ 其他 n 值($n \geq 2$)，都用遞迴呼叫計算($f(n)=f(n-1)+n$)
- 實做：

```
int f(int n){
    if (n==1) return 1;
    else return f(n-1)+n;
}
```

```
main(){
    int n;

    scanf(“%d”,&n);
    printf(“%d\n”,f(n));
}
```