# 資料結構之 C 語言重點複習（二）

## 1.指標一（基本資料型態）

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

```
---------------------------------------------
int a,*pa;float b,*pb;char c,*pc;

printf("%d %d %d\n", sizeof(pa),sizeof(pb),sizeof(pc));
---------------------------------------------
```

- 宣告及執行 int a,*pa; pa=&a;後，a、&a、&pa、pa、*pa 的意義？

```
---------------------------------------------
int a=3,*pa;
printf("%d %d %d %d %d\n",a,&a,&pa,pa,*pa);
pa=&a;
printf("%d %d %d %d %d \n",a,&a,&pa,pa,*pa);
*pa=99;
printf("%d %d %d %d %d \n",a,&a,&pa,pa,*pa);
scanf("%d",pa);
printf("%d %d %d %d %d \n",a,&a,&pa,pa,*pa);
---------------------------------------------
```

- 宣告及執行 char c,*pc; pc=&c;後，c、&c、&pc、pc、*pc 的意義？

```
---------------------------------------------
char c='W',*pc;
printf("%c %d %d %d %c\n",c,&c,&pc,pc,*pc);
pc=&c;
printf("%c %d %d %d %c\n",c,&c,&pc,pc,*pc);
*pc='Z';
printf("%c %d %d %d %c\n",c,&c,&pc,pc,*pc);
scanf("%c",pc);
printf("%c %d %d %d %c\n",c,&c,&pc,pc,*pc);
---------------------------------------------
```

## 2.指標二（字串型態）

- 宣告及執行 char c[100],*pc; pc=&c;後，c、&c、&pc、pc、*pc 的意義？

```
---------------------------------------------
char c[100]="abcd56789",*pc;
printf("%s %d %d %d %d\n",c,c,&c,&pc,pc);
```

```
pc=(char *)&c;//或 pc=(char *)c;
printf("%s %d %d %d %c %s\n",c,c,&pc,pc,*pc,pc);//pc 的%d 及%s 用法
---------------------------------------------
```

- 以指標變數操作字串(一)

```
---------------------------------------------
char c[100]=”abcd56789”,*pc;
pc=&c;

printf(“%s\n”,c);

while (*pc!=’\0’){
        printf(“%c”,*pc);
        pc++;
}
printf(“\n”);
---------------------------------------------
```

- 以指標變數操作字串(二)

```
---------------------------------------------
char c[100]=”abcd56789”,*pc;
pc=&c;

printf(“%s\n”,c);

i=0;
while (pc[i]!='\0'){
        printf("%c",pc[i]);
        i++;
}
printf("\n");
---------------------------------------------
```

2-2.指標 2-2（整數陣列以指標操作）

- 宣告及執行 int a[10],*pa; pa=(int *)a;如何以 pa 操作 a 陣列的內容？

```
---------------------------------------------
int a[10],*pa,i;
```

```
pa=(int *)a;//pa=&a;???
for (i=0;i<10;i++) a[i]=i;
for (i=0;i<10;i++) printf("%d,”,a[i]);
printf("\n”);

for (i=0;i<10;i++){
  pa[i]=i+3;//將 pa 偽裝成陣列變數
}
for (i=0;i<10;i++) printf("%d,”,a[i]);
printf("\n”);

for (i=0;i<10;i++){
  *pa=*pa+3;
    pa++;//實際編譯為 pa=pa+1*sizeof(int)
}
for (i=0;i<10;i++) printf("%d,”,a[i]);
printf("\n”);

pa=(int *)a;
for (i=0;i<10;i++){
  *(pa+i)=*(pa+i)+3; //實際編譯為*(pa+i* sizeof(int))=*(pa+i* sizeof(int))+3
}
for (i=0;i<10;i++) printf("%d,”,a[i]);
printf("\n”);
```

3.指標三（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，及指標變數 pastudent

```
---------------------------------------------
struct ASTUDENT{
        int num,score;
};
struct ASTUDENT onestudent,*pastudent;
int *pa;float *pb;char *pc;

printf(“%d %d %d %d\n”, sizeof(pa),sizeof(pb),sizeof(pc),sizeof(pastudent));
---------------------------------------------
```

- 以指標使用結構 ASTUDENT 變數 onestudent 中的成員變數

```
---------------------------------------------
pastudent=&onestudent;
printf(“%d %d\n”, &onestudent.num, &onestudent.score);
printf(“%d %d\n”, &(*pastudent).num, &(*pastudent).score);
printf(“%d %d\n”, &(pastudent->num), &(pastudent->score));

(*pastudent).num=3;
(*pastudent).score=95;
printf(“%d %d\n”, onestudent.num, onestudent.score);
printf(“%d %d\n”, (*pastudent).num, (*pastudent).score);
printf(“%d %d\n”, pastudent->num, pastudent->score);

pastudent->num=4;
pastudent->score=99;
printf(“%d %d\n”, onestudent.num, onestudent.score);
printf(“%d %d\n”, (*pastudent).num, (*pastudent).score);
printf(“%d %d\n”, pastudent->num, pastudent->score);
---------------------------------------------
```

4.指標四（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，及指標變數 pstudents

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

- 以指標使用結構 ASTUDENT 陣列變數 fourstudent 中的成員變數（一）

```
---------------------------------------------
printf("%d %d\n",fourstudent,&fourstudent);

//印出變數 fourstudent[i].num 及 fourstudent[i].score 的位址
for (i=0;i<4;i++)
```

```
    printf(“%d %d\n”, &fourstudent[i].num, &fourstudent[i].score);


//印出變數(*pstudents).num 及(*pstudents).score 的位址
printf(“-----------------\n”);
pstudents=fourstudent;//pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf(“%d %d\n”, &(*pstudents).num, &(*pstudents).score);
    pstudents++;//？？？printf(“%d\n”, pstudents);
}

//印出變數 pstudents->num 及 pstudents->score 的位址
printf(“-----------------\n”);
pstudents=fourstudent;//pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf(“%d %d\n”, &(pstudents->num), &(pstudents->score));
    pstudents++;
}

//pstudents 偽裝為陣列，印變數 pstudents[i].num 及 pstudents[i].score 位址
printf(“-----------------\n”);
pstudents=fourstudent;// pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf(“%d %d\n”, &pstudents[i].num, &pstudents[i].score);
}
```

---------------------------------------------

- 以指標使用結構 ASTUDENT 陣列變數 fourstudent 中的成員變數（二）

---------------------------------------------

```
//輸入變數 pstudents->num 及 pstudents->score 的值
pstudents=fourstudent;// pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    scanf(“%d %d”, &(pstudents->num), &(pstudents->score));
    pstudents++;
}
```

```
//以各種方式印出 struct ASTUDENT fourstudent[4]所有變數值
printf("-----------------\n");
pstudents=fourstudent;// pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf("%d %d\n", (*pstudents).num, (*pstudents).score);
    pstudents++;//？？？printf("%d\n", pstudents);
}

printf("-----------------\n");
pstudents=fourstudent;// pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf("%d %d\n", pstudents->num, pstudents->score);
    pstudents++;
}

printf("-----------------\n");
pstudents=fourstudent;// pstudents=(struct ASTUDENT *)&fourstudent;
for (i=0;i<4;i++){
    printf("%d %d\n", pstudents[i].num, pstudents[i].score);
}

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

5.指標五（動態記憶體配置）

- 宣告 int *pa，要輸入指定個數的整數

```
---------------------------------------------
int n,i,*pa;

//執行時才指定要輸入 n 個整數，動態配置剛好大小的記憶體給 pa
scanf("%d",&n);
pa=(int *)malloc(n*sizeof(int));

//第一種由 pa 讀入 n 個整數值後，再列印該 n 個整數值的做法
for (i=0;i<n;i++)
```

```
        scanf("%d",(pa+i));
for (i=0;i<n;i++)
        printf("%d\n",*(pa+i));

//第二種由 pa 讀入 n 個整數值後，再列印該 n 個整數值的做法
for (i=0;i<n;i++)
        scanf("%d",&pa[i]);
for (i=0;i<n;i++)
        printf("%d\n",pa[i]);
---------------------------------------------
```

● 宣告 char *pc，要輸入指定長度的字串

```
---------------------------------------------
int n,i;
char *pc;

//執行時才指定字串的最長字元數為 n 個字元，配置剛好的記憶體給 pc
scanf("%d",&n);
pc=(char *)malloc(n*sizeof(char));// pc=(char *)malloc(n);

scanf("%s",pc);

//第一種由 pc 印出字串內容的做法
i=0;
while (pc[i]!='\0'){
        printf("%c",pc[i]);
        i++;
}
printf("\n");

//第二種由 pc 印出字串內容的做法
printf("%s\n",pc);
---------------------------------------------
```

● 宣告 struct ASTUDENT *pstudents，要輸入指定個數的結構

```
---------------------------------------------
struct ASTUDENT{
        int num,score;
};
int n,i;
struct ASTUDENT *pstudents;

//執行時才指定要輸入 n 個結構，動態配置剛好大小的記憶體給 pstudents
scanf("%d",&n);
pstudents =( struct ASTUDENT *)malloc(n*sizeof(struct ASTUDENT));

//第一種由 pstudents 讀入 n 個結構中的變數值後，再列印內容的做法
for (i=0;i<n;i++)
      scanf("%d %d",&( pstudents +i)->num, &( pstudents +i)->score);
for (i=0;i<n;i++)
      printf("%d %d\n",( pstudents +i)->num, (pstudents +i)->score);

//第二種由 pstudents 讀入 n 個結構中的變數值後，再列印內容的做法
for (i=0;i<n;i++)
      scanf("%d %d",& pstudents [i].num, & pstudents [i].score);
for (i=0;i<n;i++)
      printf("%d %d\n", pstudents [i].num, pstudents [i].score);
---------------------------------------------
```