พอยน์เตอร์

ประโยชน์ของการใช้ พอยน์เตอร์

1.ใช้รับค่า address ของตัวแปรชุด (arrays)หรือตัวแปรสตริง จากฟังก์ชั่นหนึ่งไปอีกฟังก์ชันหนึ่ง และใช้ในการส่งค่ากลับมายังฟังก์ชั่น

2. ใช้ในการจัดการตัวแปรชุดหรือตัวแปรสตริงให้มีประสิทธิภาพ โดยการอ้างอิงตำแหน่งของตัวแปรชุดหรือตัวแปรสตริงโดยตรง ทำให้สามารถจัดเก็บและเรียกใช้ข้อมูลที่ต้องการได้ง่าย

3.ใช้ในการจัดโครงสร้างข้อมูล

ความหมายของพอยน์เตอร์

พอยน์เตอร์ คือ ตัวแปรชนิดหนึ่งที่เก็บตำแหน่ง ของข้อมูลภายในหน่วยความจำ ซึ่งการเก็บตำแหน่งจะเก็บเฉพาะตำแหน่งแรกของข้อมูลเท่านั้น

ประเภทของพอยน์เตอร์

1. Direct pointer ตัวแปรที่เก็บตำแหน่งแรกของข้อมูลถายในหน่วยความจำโดยตรง

2. Indirect pointer คือตัวแปรที่เก็บตำแหน่งของพอยน์เตอร์อีกตัวหนึ่ง บางครั้งเรียก pointer to pointer

การประกาศตัวแปรพอยน์เตอร์

Type *ptr_name; กรณีที่มีพอยเตอร์ 1 ตัว

หรือ

Type *ptr1_name,*ptr2_name,…; กรณีที่มีพอยเตอร์มากกว่า 1 ตัว

type คือ ชนิดตัวแปรพอยน์เตอร์เป็นชนิดเดียวกันกับข้อมูล

ptr_name คือ ชื่อตัวแปรพอยเตอร์ โดยจะต้องมีเครื่องหมาย *(asterisk) นำหน้าชื่อเพื่อบอกให้ compiler รู้ว่าเป็นตัวแปรพอยเตอร์

การประกาศตัวแปรพอยน์เตอร์

Type *ptr_name; กรณีที่มีพอยเตอร์ 1 ตัว

หรือ

Type *ptr1_name,*ptr2_name,…; กรณีที่มีพอยเตอร์มากกว่า 1 ตัว

type คือ ชนิดตัวแปรพอยน์เตอร์เป็นชนิดเดียวกันกับข้อมูล

ptr_name คือ ชื่อตัวแปรพอยเตอร์ โดยจะต้องมีเครื่องหมาย *(asterisk) นำหน้าชื่อเพื่อบอกให้ compiler รู้ว่าเป็นตัวแปรพอยเตอร์

ptr1_name, ptr2_name,… คือ ชื่อตัวแปรพอยเตอร์ 1, 2 ตามลำดับ

ตัวอย่างการประกาศตัวแปรพอยน์เตอร์

int *ptr; กรณีที่มีพอยเตอร์ 1 ตัว

Float *pone,*ptwo, *pthree

Char *ptrx, *ptry, *ptrz;

การกำหนดค่าตำแหน่งข้อมูลให้ตัวแปรพอยน์เตอร์

รูปแบบการกำหนดค่า address

ptr_name=&variable_name;

ptr_name คือ ชื่อตัวแปรพอยน์เตอร์

variable_name;ชื่อตัวแปรที่ต้องการกำหนดค่า address แรกให้ตัวแปรพอยน์เตอร์

ในกรณีเป็นตัวแปรชุด หรือตัวแปรสตริงไม่ต้องใส่เครื่องหมาย & นำหน้า

เช่น char s1[80]="computer"; char *ptrs1;
ptrs1=s1;

ตัวดำเนินการที่ใช้กับพอยน์เตอร์

ตัวดำเนินการ * **(asterisk)** เป็นเครื่องหมายที่บอกค่าข้อมูลที่พอยน์เตอร์ชี้อยู่

```
/* asterisk.c writer: */
#include <stdio.h>
void main (void){
     int x=5,y=7;
     int *px, *py;
    px=&x;
     py=&y;
     printf("x=%d,y=%d",*px,*py);
}
```

ตัวดำเนินการ **& (ampersand)** เป็นเครื่องหมายที่บอกค่า **address** ของข้อมูล

```
#include <stdio.h>
    void main (void){
        int a=10,b=20;
        int *pa, *pb;
        pa=&a;
        pb=&b;
    printf("A1=%d,B1=%d\n",a,b);
    printf("A2=%d,B2=%d\n",*pa,*pb);
    printf("Address of a,b are %p\t %p
    \n",&pa,&pb);
    }
```

พอยเตอร์กับฟังก์ชันที่เขียนขึ้นใช้งานเอง

(pointer and user defined functions)

การส่งค่าไปยังฟังก์ชัน สามารถทำได้ 2 วิธี

1. การส่งค่าข้อมูลไปยังฟังก์ชันที่ต้องการ คือการคัดลอกข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง

2. การส่งตำแหน่งของข้อมูลไปยังฟังก์ชัน

พอยเตอร์กับฟังก์ชันที่เขียนขึ้นใช้งานเอง

(pointer and user defined functions)

การส่งค่าไปยังฟังก์ชัน สามารถทำได้ 2 วิธี

1. การส่งค่าข้อมูลไปยังฟังก์ชันที่ต้องการ คือการคัดลอกข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง

2. การส่งตำแหน่งของข้อมูลไปยังฟังก์ชัน

พอยเตอร์กับฟังก์ชันที่เขียนขึ้นใช้งานเอง

(pointer and user defined functions)

```
#include<stdio.h>
void f1 (int,int);
void f2 (int*,int*);
void main(void){
    int p=10, q=20;
    f1(p,q);
    f2(&p,&q);
    }
    void f1(int pp,int qq){
        printf("P1=%d\tQ1=%d\n",pp,qq);
        }
            void f2(int *pp,int *qq){
          *pp=*pp+5; *qq=*qq+10;
        printf("P2=%d\tQ2=%d\n",*pp,*qq);
}
```

การคืนค่ากลับของฟังก์ชันที่เขียนขึ้นใช้งานเอง

(Return data from functions)

```
#include<stdio.h>
void back (int*,int*);
void main(void){
    int p, q;
    back(&p,&q);
printf("P=%d, Q=%d\n",p,q);
    }
    void back(int *pp,int *qq)
{
      *pp=100;
      *qq=200;
}
```

อาร์เรย์ของพอยเตอร์

(array of pointer)

Type *ptrname[size];

หรือ type *ptrname[size]={valuelist};

ตัวอย่างการประกาศตัวแปรชุดของพอยน์เตอร์

int *px[10];

```
#include<stdio.h>
void main(void){
    int j;
    int *ptr[5];
int y[5] = {100,200,300,400,500};
for(j=0; j<5; j++)
      ptr[j]=&y[j]
for(j=0; j<5; j++){
      *ptr[j]= *ptr[j]+50
printf("y[%d]=%d\n",j,*ptr[j]);
}
}
```

pointer to pointer

type **pptrname;

หรือ

type **pptrname1,pptrname2;**

```
#include<stdio.h>
void main(void){
    int x=20,y=30;
    int *px,*py;
int **ppx,**ppy;
      px=&x;  py=&y;
      ppx=&px;        ppy=&py;
      printf("X=%d,Y=%d\n",**ppx,**ppy);
}
```

การบ้าน

1 จงอธิบายความแตกต่างระหว่างพอยน์เตอร์กับตัวแปรว่าแตกต่างกันอย่างไร

2 จงยกตัวอย่างชนิดของตัวแปรพอยน์เตอร์ และการประกาศพอยน์เตอร์ , อาร์เรย์พอยน์เตอร์ ?

3 การกำหนดค่าข้อมูลโดยใช้พอยน์เตอร์อ้างอิงทำได้อย่างไร และมีการทำงานอย่างไร ?

4 จงเขียนโปรแกรมโดยใช้พอยน์เตอร์ในการอ้างอิงถึงข้อมูลในตัวแปรที่เป็นอาร์เรย์ กำหนดให้มีอาร์เรย์แบบ char ชื่อ sample[] เก็บตัวอักษรเรียงติดกัน คือ abcddcbaabccbaabbaaa แล้วให้รับตัวอักษรจากแป้นพิมพ์ 1 ตัวอักษร และนำไปตรวจสอบกับตัวอักษรในอาร์เรย์ sample[ ] ว่าเหมือนกันที่ตำแหน่งใด และเหมือนทั้งหมดกี่ตัวอักษร ?

การบ้าน

1 จงอธิบายความแตกต่างระหว่างพอยน์เตอร์กับตัวแปรว่าแตกต่างกันอย่างไร

2 จงยกตัวอย่างชนิดของตัวแปรพอยน์เตอร์ และการประกาศพอยน์เตอร์ , อาร์เรย์พอยน์เตอร์ ?

3 การกำหนดค่าข้อมูลโดยใช้พอยน์เตอร์อ้างอิงทำได้อย่างไร และมีการทำงานอย่างไร ?

4 จงเขียนโปรแกรมโดยใช้พอยน์เตอร์ในการอ้างอิงถึงข้อมูลในตัวแปรที่เป็นอาร์เรย์ กำหนดให้มีอาร์เรย์แบบ char ชื่อ sample[] เก็บตัวอักษรเรียงติดกัน คือ abcddcbaabccbaabbaaa แล้วให้รับตัวอักษรจากแป้นพิมพ์ 1 ตัวอักษร และนำไปตรวจสอบกับตัวอักษรในอาร์เรย์ sample[ ] ว่าเหมือนกันที่ตำแหน่งใด และเหมือนทั้งหมดกี่ตัวอักษร ?