知识点总结

·在C语言中，指针是用于储存变量地址的变量。
·定义一个指针的语法格式：数据类型* 指针名;
其中，数据类型指的是，指针所指向的变量的数据类型；
后面的*是一个说明符，用来说明该变量是指针。
例：int* ptr; 定义了一个指向整数的指针ptr。

·星号 * 可以放在变量名与类型关键字之间的任何地方：
数据类型 * 指针名;
数据类型 *指针名;

·通过取址符（又叫地址运算符）&来获取变量的地址。
语法格式：&变量名
例：将变量var的地址赋给指针ptr。

```
int var = 10;
int* ptr = &var;
```

·使用间址运算符（又叫解引用运算符）*来通过指针访问或修改其所指向变量的值。

语法格式：*指针名

例：通过指针ptr访问var的值，并把该值赋给变量value，然后通过指针ptr更新var的值。

```
int var = 10;

// 定义语句里的*是一个说明符，用来说明该变量是指针

int* ptr = &var;

// 下面两条语句里的*是间址运算符

int value = *ptr;

*ptr = 20;
```

·注意：

1、所有类型的指针变量在同一系统中所占的内存大小是相同的，因为指针储存的是地址，地址的大小在系统里是固定的。但在不同的系统里，地址的大小不同。在32位系统里，地址一般占4个字节；在64位系统里，地址一般占8个字节。

2、如果暂时不知道要给它赋哪个变量的地址，可以先给它赋一个NULL值（NULL被定义在stdio库中，所以使用NULL前，需要包含stdio.h头文件）。这样表明该指针是空指针，不指向任何一个有效的地址。不然，未初始化的指针会指向一个不确定的位置，给程序运行造成未知风险。

*星号是一个说明符，用来说明这个变量是指针

题目1、若有定义语句：double a, *p = &a;

以下叙述中错误的是（C）

A．定义语句中的p只能存放double类型变量的地址

B．定义语句中的*号是一个说明符

C．定义语句中的*号是一个间址运算符

D．定义语句中*p = &a，把变量a的地址作为初值赋给指针变量p

题目2、设已有定义：`float x;`

则以下对指针变量p进行定义且赋初值的语句中正确的是（A）

A．float *p = &x;

B．int *p = (int)x;

C．float p = &x;

D．float *p = 1024;

题目3、有以下程序

```
#include <stdio.h>
main()
{
    int a = 1, b = 3, c = 5;
    int *p1 = &a, *p2 = &b, *p = &c;
    *p = *p1 * (*p2);
    printf("%d\n", c);
}
```

执行后的输出结果是（D）

A. 4

B. 2

C. 1

D. 3

题目4、有如下程序:

```
#include <stdio.h>
main()
{
    int a = 0, *ptr;
    ptr = &a;
    *ptr = 3;
    a = (*ptr)++;
    printf("%d,%d\n", a, *ptr);
}
```

程序运行后的输出结果是（A）

A. 4,4

B. 0,1

C. 1,4

D. 0,4

题目5、有以下程序:

```
#include <stdio.h>
main(){
    int m = 1, n = 2, *p = &m, *q = &n, *r;
    r = p; p = q; q = r;
    printf("%d,%d,%d,%d\n", m, n, *p, *q);
}
```

相当于交换了指针p,q的值（即指向对象）

程序运行后的输出结果是（B）

A. 1,2,1,2

B. 1,2,2,1

C. 2,1,2,1

D. 2,1,1,2