知识点总结

·在C语言中，宏定义是一种编译预处理指令，它允许我们用一个标识符（宏名）来替换一段文本。

·宏定义分为两种类型：不带参数的宏和带参数的宏。

·不带参数的宏：

定义的语法格式：#define 宏名 替换文本

·注意：

1、宏定义不是语句，在行末不用加分号。如果加上分号，则连分号也一起替换。

2、在C程序中，宏的定义位置一般写在程序的开头。

3、宏名一般全用大写字母表示，便于与变量名区别，但这不是强制规定。

·例：

```c
#include <stdio.h>
// 宏名：PI，替换文本：3.14159
#define PI 3.14159

int main() {
    double radius = 5.0;
    // 此处的PI在预处理时被替换为3.14159
    double area = PI * radius * radius;
    printf("The area of the circle is: %lf\n", area);
    return 0;
}
```

在预处理阶段，预处理器会将源程序中所有出现宏名的地方替换为对应的替换文本，
这只是一种简单的代换，预处理程序对它不做任何检查。

·用宏定义常量与用const关键字定义常量的区别：

1、类型检查：宏定义仅简单文本替换，无类型检查，可能致类型错误；const常量有明确数据类型，编译器会做类型检查，安全性更高。

2、作用域：宏定义常量作用域从定义处起，至文件结束或遇到#undef指令取消定义为止；const常量的作用域与普通变量相同，例：函数内定义的const常量，其作用域仅限于该函数。

·带参数的宏：

定义的语法格式：#define 宏名(参数1, 参数2, ...) 替换文本

例：

```
#include <stdio.h>
// 宏名：MAX，参数：a和b，替换文本：((a) > (b)? (a) : (b))
#define MAX(a, b) ((a) > (b)? (a) : (b))

int main() {
    int num_1 = 10;
    int num_2 = 20;
    // 此处的MAX(num_1, num_2)，预处理器会将其替换为((num_1) > (num_2)?
(num_1) : (num_2))
    int max_value = MAX(num_1, num_2); ((num_1)>(num_2)?(num_1):(num_2))
    printf("The maximum value is: %d\n", max_value);
    return 0;
}
```

·带参数的宏与函数的区别：

1、函数调用在运行时执行，有参数传递、返回值处理等开销，会占用运行时间；而宏替换在预处理阶段进行，没有这些开销，不占用运行时间。因此，对于频繁调用且代码简短的操作，使用宏可能会提高效率。

2、函数的参数有明确的类型检查，而宏参数只是简单的文本替换，不会进行类型检查。这可能导致宏在使用不当时，产生难以发现的错误。

·注意：

1. 对于带参数的宏，为了避免宏替换可能产生的运算符优先级问题，导致得到错误的结果，替换文本中的参数和整个表达式都应该用括号括起来。

例如：在 `#define MAX(a, b) ((a) > (b)? (a) : (b))` 中，

如果不写括号：`#define MAX(a, b) a > b? a : b` ，

遇到像 `int result = MAX(3, 2) * 5;` 这样的调用时，文本替换后是这样的：3 > 2 ? 3 : 2 * 5 ，

此时，`int result = MAX(3, 2) * 5;` 这条语句的实际执行逻辑变为：如果3大于2，就给result赋值为3，否则就给result赋值为10。

这与我们原本期望的“取3和2中的较大值，再与5相乘，并将计算结果赋值给result”的逻辑不符。

2. 宏定义的作用域：宏定义从定义点开始到源文件结束都有效，除非用#undef指令取消定义。

#undef指令取消宏定义的语法格式：#undef 宏名

例：

```c
#include <stdio.h>

#define VALUE 10

int main() {
    printf("VALUE is: %d\n", VALUE);
    // 用#undef指令取消宏VALUE的定义
    #undef VALUE
    // 在此处VALUE已被取消定义，再次使用会导致编译报错
    printf("VALUE is: %d\n", VALUE);
    return 0;
}
```

宏替换不具有计算功能，宏是简单的文本替换，不占用运行时间
宏是预处理指令的一种
宏名可以是任何合法的标识符

题目1、以下关于宏的叙述错误的是（ C ）

A．宏替换不具有计算功能

B．宏是一种预处理指令

C．宏名必须用大写字母构成

D．宏替换不占用运行时间

题目2、有以下程序:

```
#include <stdio.h>
#define S(x) x * x
main()
{
    int k = 5, j = 2;
    printf("%d,%d\n", S(k + j + 2), S(j + k + 2));
}
```

程序的运行结果是（A）

A. 21,18

B. 81,81

C. 21,21

D. 18,18

题目3、有以下程序:

```
#include <stdio.h>
#define f(x) x*x*x
main()
{
    int a = 3, s, t;
    s = f(a + 1);
    t = f((a + 1));
    printf("%d,%d\n", s, t);
}
```

程序运行后的输出结果是（A）

A. 10,64

B. 10,10

C. 64,10

D. 64,64

题目4、有以下程序:

```
#include <stdio.h>
#define PT 3.5
#define S(x) PT * x * x
main()
{
    int a = 1, b = 2;
    printf("%4.1f\n", S(a + b));
}
```

程序运行后的输出结果是（C）

A. 14.0

B. 31.5

C. 7.5

D. 程序有错无输出结果

题目5、有以下程序:

```
#include <stdio.h>
#define M 5
#define f(x, y) x * y + M

main()
{
    int k;
    k = f(2, 3) * f(2, 3);
    printf("%d\n", k);
}
```

程序的运行结果是（B）

A. 22

B. 41

C. 100

D. 121