



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计

Programming in C++



1011018

主讲：魏英，计算机学院

结构化程序设计习题

- ◆ 4、预处理命令
- ◆ 5、数组
- ◆ 6、指针与引用
- ◆ 7、自定义数据类型

24.4 预处理命令

1. 以下叙述中错误的是__C__

- A. 预处理命令都必须以#开头
- B. 在程序中凡是以#开始的语句行都是预处理命令行
- C. C++程序在执行过程中对预处理命令行进行处理
- D. 预处理命令行可以出现在C++程序中任意一行上

2. 设#define P(x) x/x, 执行语句cout<<P(3*5)后的输出结果是__C__

- A. 1
- B. 0
- C. 25
- D. 15

24.4 预处理命令

3. 在“文件包含”预处理命令形式中，当#include后面的文件名用“”扩起来时，寻找被包含文件的方式是 **B**

- A. 直接按系统设定的标准方式搜索目录
- B. 现在源程序所在的目录中搜索，再按系统设定的标准方式搜索
- C. 仅仅搜索源程序所在的目录
- D. 仅仅搜索当前目录

4. 若有宏定义：

```
#define N 3
```

```
#define Y(n) ((N+1)*n)
```

则表达式 $2*(N+Y(5+1))$ 的值是 **C**

- A. 出错
- B. 42
- C. 48
- D. 54

24.5 数组

1. 下列关于数组的描述正确的是 **B**

- A. 数组的长度是固定的，而其中元素的数据类型可以不同
- B. 数组的长度是固定的，而其中元素的数据类型必须相同
- C. 数组的长度是可变的，而其中元素的数据类型可以不同
- D. 数组的长度是可变的，而其中元素的数据类型必须相同

2. 在C++语言中引用数组元素时，下面关于数组下标数据类型的说法错误的是 **D**

- A. 整形常量
- B. 整形表达式
- C. 整形常量或整形表达式
- D. 任何类型的表达式

3. 下面二维数组的定义中正确的是 **C**

A. `int a[][]={1,2,3,4,5,6};`

B. `int a[2][]={1,2,3,4,5,6};`

C. `int a[][3]={1,2,3,4,5,6};`

D. `int a[2,3]={1,2,3,4,5,6};`

4. 下面有关字符数组的描述错误的是 **C**

A. 字符数组可以存放字符串

B. 字符串可以整体进行输入和输出

C. 可以在赋值语句中通过赋值运算符对字符数组整体赋值

D. 不可以用关系运算符对字符数组中的字符串进行比较

24.5 数组

5. 字符数组s不能作为字符串使用的是 **D**

A. `char s[]="happy";`

B. `char s[6]={'h','a','p','p','y','\0'};`

C. `char s[]={"happy"};`

D. `char s[5]={'h','a','p','p','y'};`

6. 下面程序的输出结果是 **B**

```
char str[]="SSWLIA",c; int k;
for(k=2;(c=str[k])!='\0';k++){
    switch(c){
        case 'I': ++k; break;
        case 'L': continue;
        default : cout<<c; continue;
    }
    cout<<'*';
}
```

A. SSW

B. SW*

C. SW*A

D. SW

7. 对数组名作为函数的参数，下面的描述正确的是 **B**

- A. 数组名做函数的参数，调用时将实参数组复制给形参数组
- B. 数组名做函数的参数，主调函数和被调函数共用一段存储空间
- C. 数组名做函数的参数，形参定义的数组长度不能省略
- D. 数组名做函数的参数，不能改变主调函数中的数据

24.6 指针与引用

1. 下面对指针的描述不正确的是 D

- A. 指针是地址变量
- B. 指针不能用除0以外的常量赋值
- C. 两个指针变量的加法无意义
- D. 指向不同类型变量的指针的长度也不同

2. 变量的指针，其含义是指该变量的 B

- A. 值
- B. 地址
- C. 名
- D. 一个标志

3. 对于类型相同的两个指针变量之间不能进行的运算是 C

- A. <
- B. =
- C. +
- D. -

24.6 指针与引用

4. 若有定义`int x[10]={0,1,2,3,4,5,6,7,8,9},*p1;`，则数值不为3的表达式是 C

A. `x[3]`

B. `p1=x+3,*p1++`

C. `p1=x+2,*(p1++)`

D. `p1=x+2,*++p1`

5. 若有定义`char ch[]="abc\0def",*p=ch;`，执行语句`cout<<*p+4;`后的输出结果是 C

A. `def`

B. `d`

C. `e`

D. `0`

6. 若有定义`int(*p)();`，标识符`p`可以__**C**__

A. 表示函数的返回值

B. 指向函数的入口地址

C. 表示函数的返回类型

D. 表示函数名

7. 下列引用的定义中，__**B**__是错误的。

A. `int i;`
`int &j=i;`

B. `int i;`
`int &j;`
`j=i;`

C. `float i;`
`float &j=i;`

D. `char d;`
`char &k=d;`

24.7 自定义数据类型

1. 设有结构体说明`struct ex{int x; float y; char z;}example;`，以下叙述中错误的是 **B**

- A. `struct`是结构体类型的关键字
- B. `example`是结构体类型名
- C. `x`、`y`、`z`都是结构体成员名
- D. `struct ex`是结构体类型

2. 在说明一个结构体变量时，系统分配给它的存储空间是 **D**

- A. 该结构体中的第一个成员所需的存储空间
- B. 该结构体中的最后一个成员所需的存储空间
- C. 该结构体中占用最大存储空间的成员所需的存储空间
- D. 该结构体中所有成员所需的存储空间之和

24.7 自定义数据类型

3. 若有定义struct ru{ long x; float y; }time,*timep=&time;, 则对time中成员x的正确引用是 **C**

- A. ru.time.x B. timep.x C. (*timep).x D. time->x

4. 若有定义union data { char ch; int x; }a;, 下列语句中 **A** 是不正确的。

- A. a={'x',10}; B. a.x=10; a.x++;
C. a.ch='x'; a.ch++; D. a.x=10; a.ch='x';

24.7 自定义数据类型

5. 若有定义enum color{ red,yellow=2,blue,white,black}r=white;, 执行cout<<r;后的输出结果是 **D**

- A. 0 B. 1 C. 3 D. 4

6. 下面对typedef的叙述中错误的是 **B**

- A. 用typedef可以定义各种类型名，但不能用来定义变量
- B. 用typedef可以增加新类型
- C. 用typedef只是将已存在的类型用一个新的标识符来代表
- D. 使用typedef有利于程序的通用和移植

CP 程序设计