# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

注：直接在本文件上作答，画出程序执行过程的内存变化即可

★ 首先画出三句定义语句结束后内存中各变量的所占空间及初值

★ 每个执行语句的每一步执行完成后的内存中各变量的所占空间及值

★ 每步变化一个页面(例：**++pp，分三步计算，需要三页)

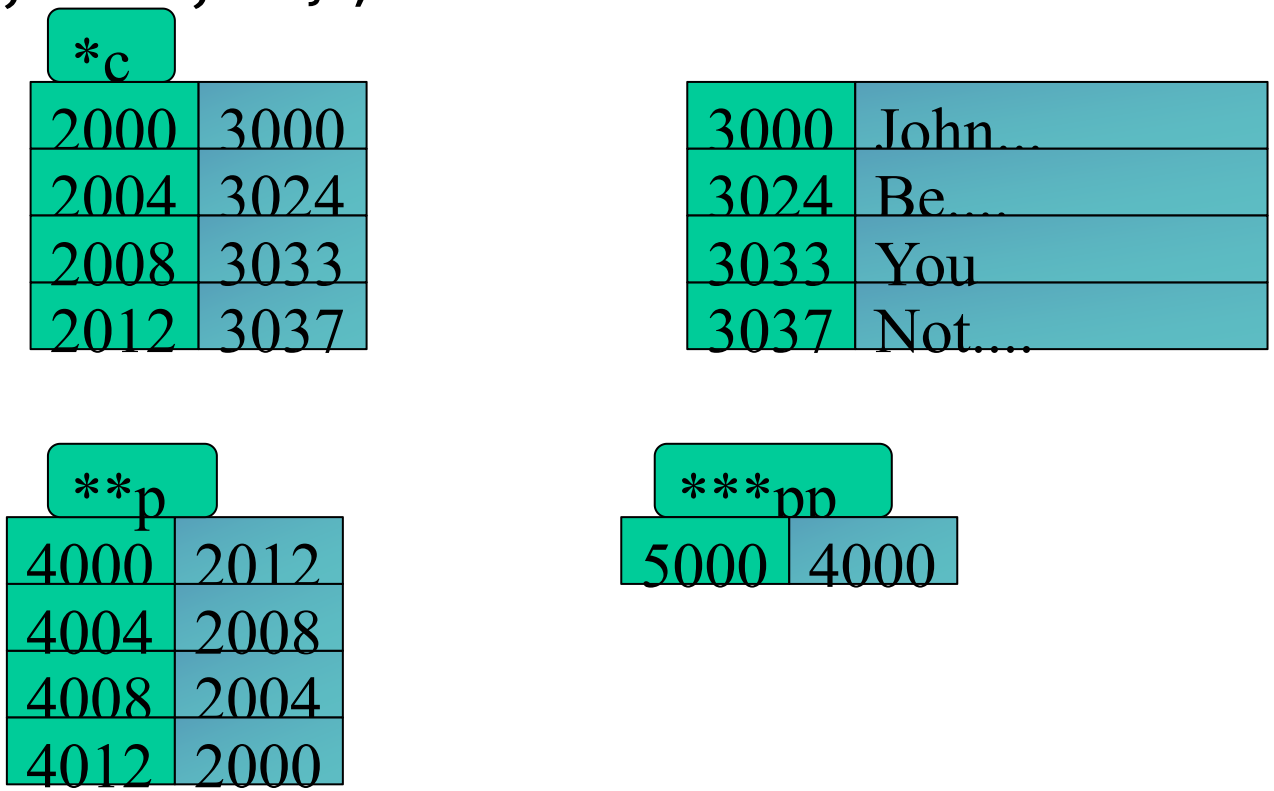
★ 不允许手写在纸上，再拍照贴图

★ 允许在各种软件工具上完成，再截图贴图

★ 转换为pdf后提交

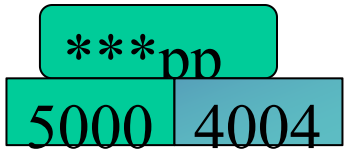
# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```



# 写出下列程序的运行结果

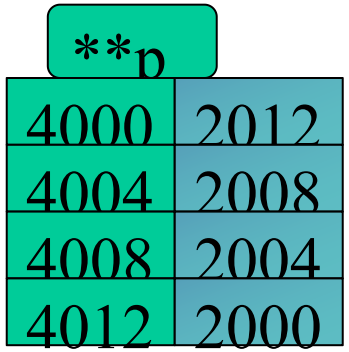
```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

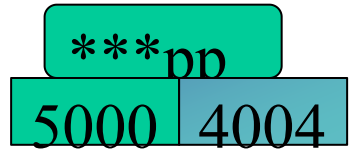



++pp:pp指向了p[1]

# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

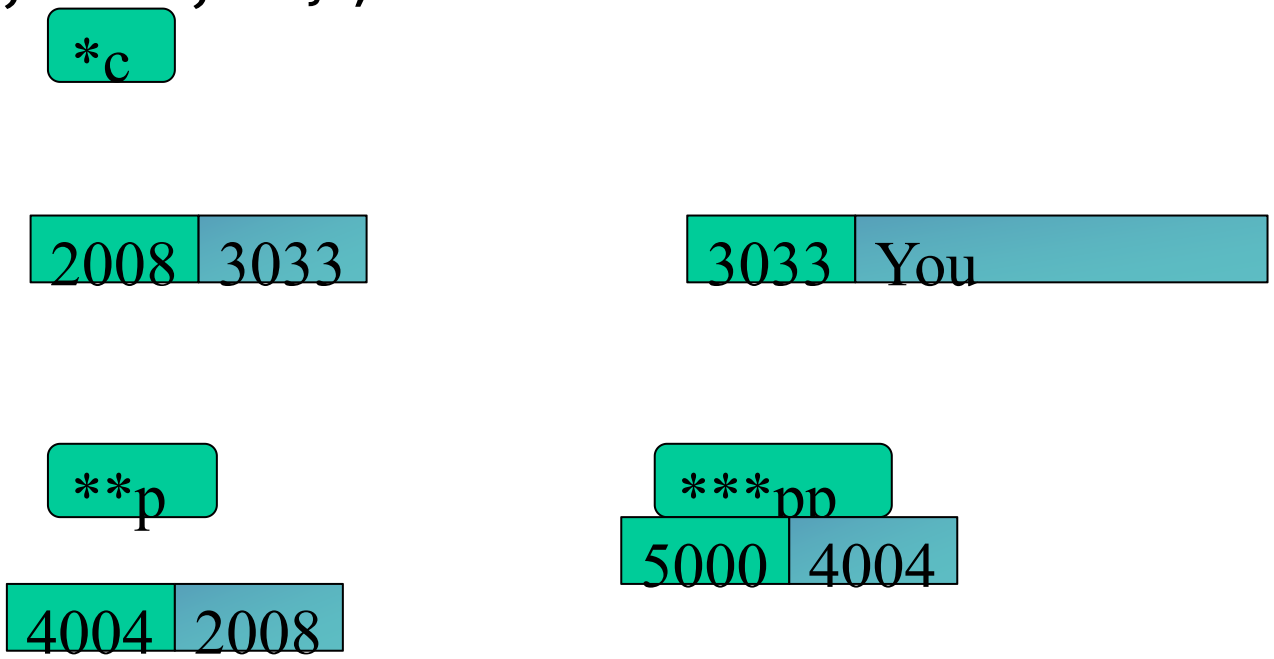



*++pp:取到了地址2008

# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

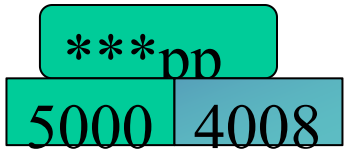



**++pp:取到了地址3033

输出：You

# 写出下列程序的运行结果

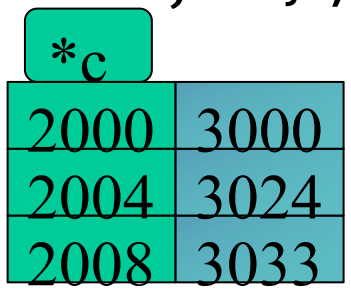
```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

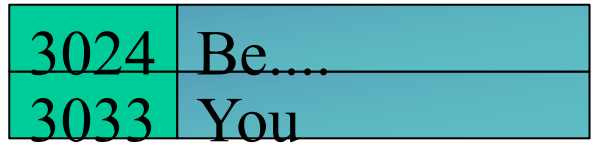



++pp:pp指向了p[1]

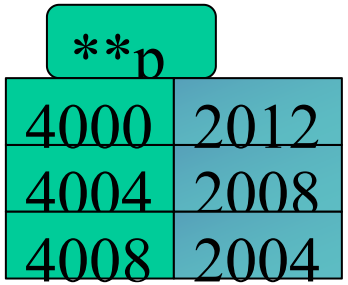
# 写出下列程序的运行结果

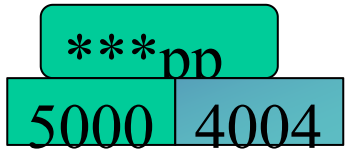
```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```



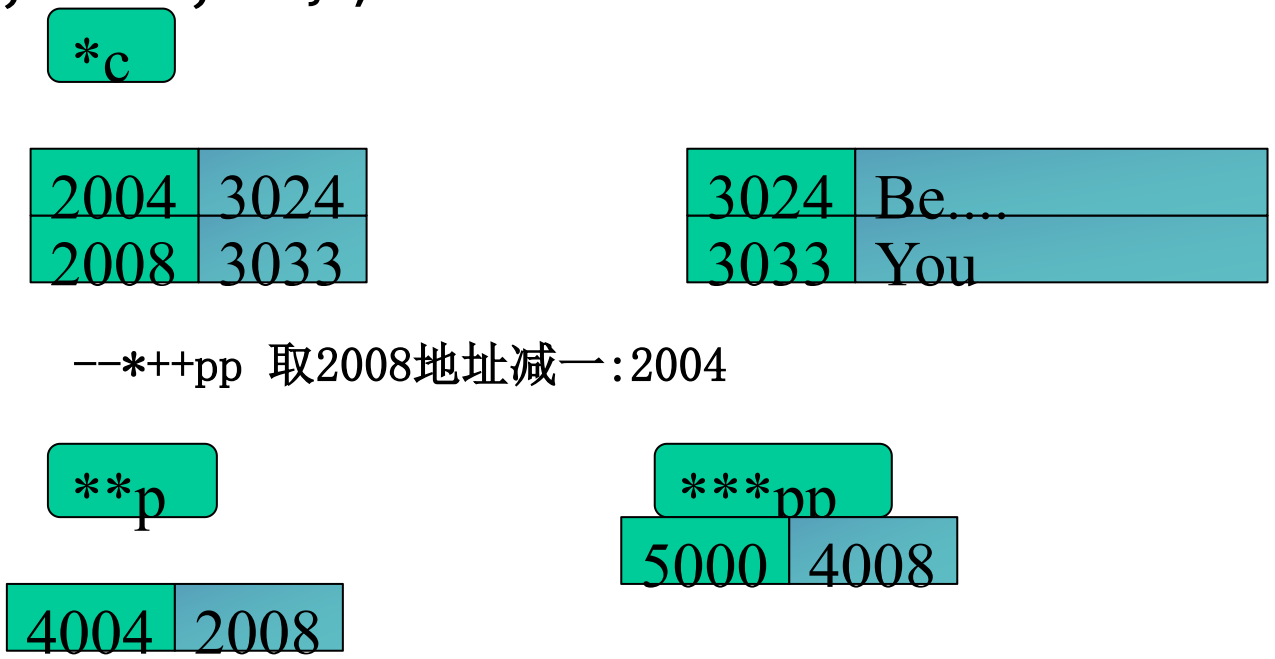

*++pp 取值p[1]:2008



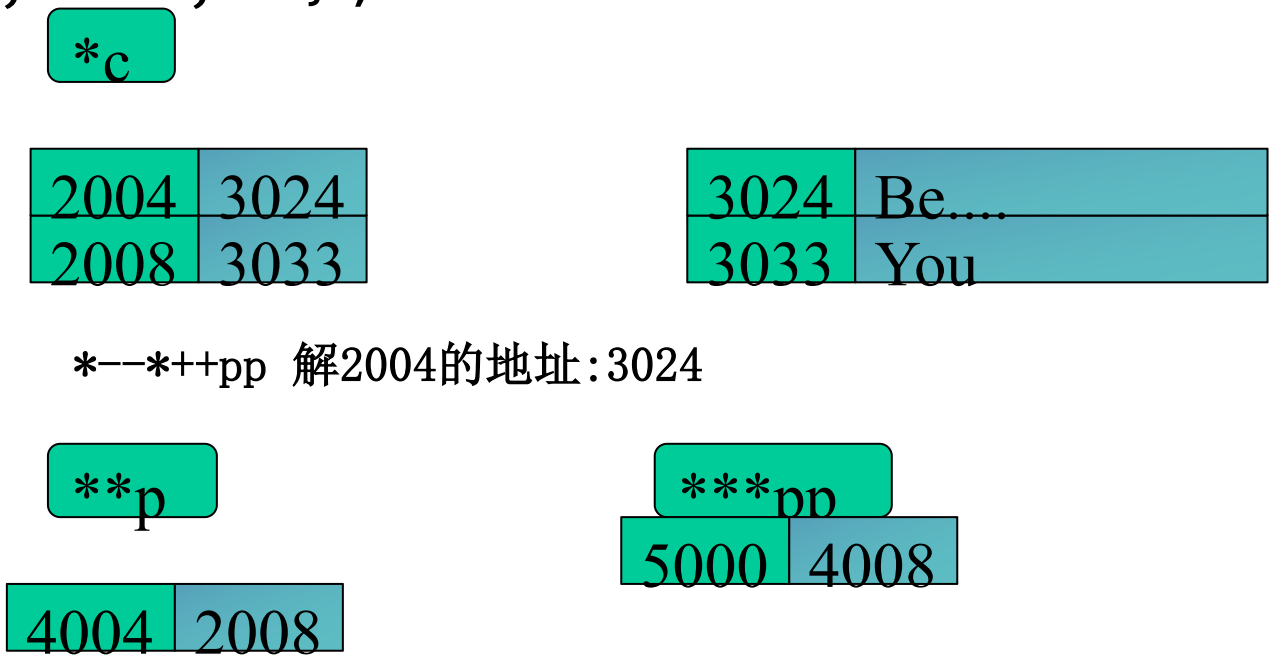
# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```



# 写出下列程序的运行结果

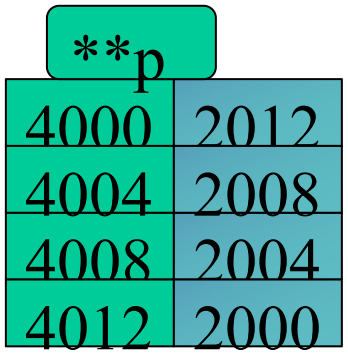
```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

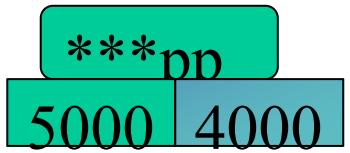

*--*++pp+4 从第四个开始输出：
ell!

# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

pp[-2]:取到了p向后两个基类型的地址 3992

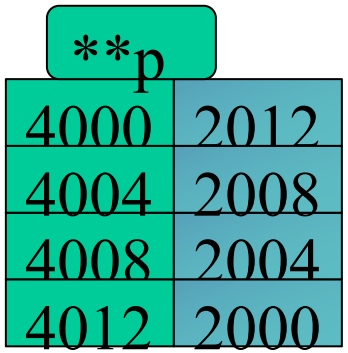


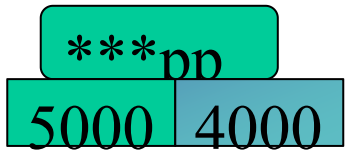



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

*pp[-2]:取到了p向后两个基类型的地址的值 --“未知”

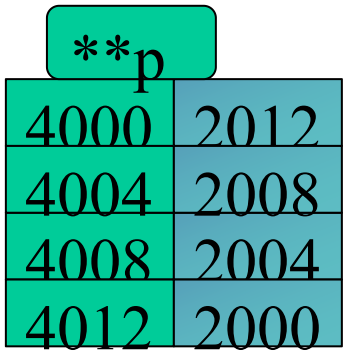


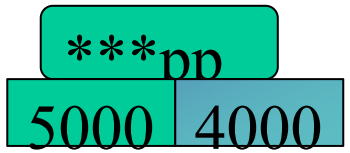



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

*pp[-2]+3:未定义行为输出，返回-192341523

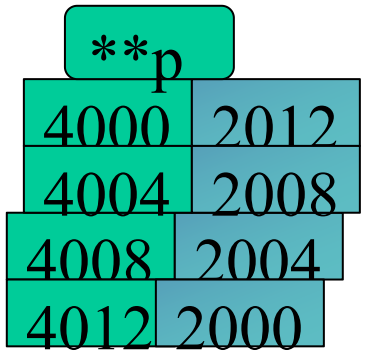


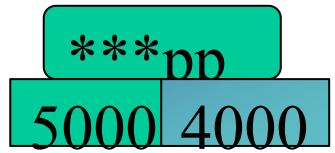



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

pp[-1]:取到了p向后1个基类型的地址的内容

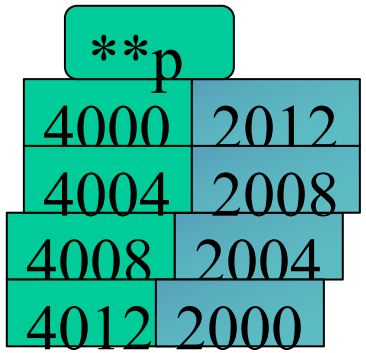


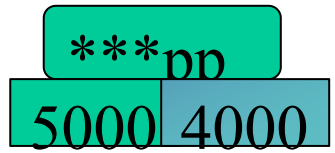



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

pp[-1][-1]:取到了p向后1个基类型的地址的内容的向后一个基类型的内容



# 写出下列程序的运行结果

```
int main()
{  const char *c[]={"John learn C++ language",
                    "Be well!", "You", "Not very"};
   const char **p[]={c+3, c+2, c+1, c};
   const char ***pp=p;
   cout << (**++pp);
   cout << (*--*++pp+4);
   cout << (*pp[-2]+3);
   cout << (pp[-1][-1]+2);
   cout << endl;
   return 0;
}
```

pp[-1][-1]+2:取到了p向后1个基类型的地址的内容的向后一个基类型的内容再向后两个基类型

未定义操作

返回-1073741819