## 6-b1

#### 源程序代码

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

#### 图例

内存单元的值

内存单元的值(指针变量)

变量名

内存地址号码

内存单元的值(在本次操作中被修改)

内存单元的值(临时调整参与运算,原值不变)



指向(指针运算符\*操作方向)

#### 内存分析

J	О	h	n	\32	1	e	•••	a	n	g	u	a	g	e	\0
1000	1001	1002	1003	1004	1005	1006	•••	1017	1018	1019	1020	1021	1022	1023	1024
В	е	132	w	е	1	1	!	!	\0						
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034						
Y	0	6	\0												
1035	1036	1037	1038										串常		
N	0	-	\32	v	е	r	у	72					存作		
1039	1040	1041	1042	1043	1044	1045	1046	1047			尼兴	<b>以</b> ,	不自	它与/	$\bigwedge_{\circ}$
	`	<u></u>									\				
					<u></u>	_			\		\	c[0]	100	00 20	000
								<u></u>			\	c[1]	7 102	25 20	004
										_	<u></u>	c[2]	7 103	35 20	800
	指针图	变量c[	i] (1紫	及)								c/3	7 103	39 20	012

基类型: char型 基类型占空间: 1字节 值: -----

(实际上不存在,只是一种理解方式)

#### 2000 c/071000 c/171025 2004 2008 c/271035 c/371039 2012

#### 内存分析

C

2000

2100

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

值: 2000 (从此内存空间开始 的4个字节被分配给用户使用)

指针变量c[i](1级)

基类型: char型

基类型占空间: 1字节

值: -----

(实际上不存在, 只是一 种理解方式)

2200 p[0]2012 2008 2204 p[1]p/272004 2208 p/372000 2212

2200 2300 p

指针变量p(3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

值: 2200 (从此内存空间开始 的4个字节被分配给用户使用)

2400 2200

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间: 4字节

值: 2200

指针变量p[i](2级)

基类型: 指向char型指针

基类型占空间: 4字节

值: -----

(实际上不存在,只是一 种理解方式)

cout << (\*\*++pp);

c 2000 2100

指针变量c[i](1级)

基类型: char型

基类型占空间: 1字节

值: -----

(实际上不存在,只是一种理解方式)

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

值:2000(从此内存空间开始的4个字节被分配给用户使用)

 p[0]
 2012
 2200

 p[1]
 2008
 2204

 p[2]
 2004
 2208

 p[3]
 2000
 2212

p 2200 2300

指针变量p[i](2级)

基类型: 指向char型指针

基类型占空间: 4字节

值: -----

(实际上不存在,只是一种理解方式)

指针变量p (3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

值: 2200 (从此内存空间开始的4个字节被分配给用户使用)

pp 2200 2400

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间:4字节

值: 2200

#### c[0] 1000 2000 c/1/1025 2004 c/21035 2008 c/3/1039 2012

#### 等价: cout << (\*(\*(pp=pp+1)));

2100 2000

指针变量c(2级)

基类型:指向char型指针

基类型占空间: 4字节

\_\_\_\_\_\_指针变量c[i](**1**级)

基类型: char型

基类型占空间: 1字节

(实际上不存在, 只是一 种理解方式)

p[0]2012 2200 2008 2204 p[1]p/272004 2208 p/372000 2212

2300 2200 р

指针变量p(3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

指针变量p[i](2级)

基类型: 指向char型指针 基类型占空间: 4字节

(实际上不存在, 只是一 种理解方式)

pp\_\_\_2204 2400

指针变量pp (3级) 基类型:指向'指向 char型指针'的指针 基类型占空间: 4字节

#### 等价: cout << (\*(\*(pp=pp+1));

	_								
J	О	h	n	\32	1	e	•••	a	n
1000	1001	1002	1003	1004	1005	1006	•••	1017	1018
В	е	\32	w	е	1	1	!	!	\0
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034
Y	О	u	\0						
1035	1036	1037	1038						
N	0	t	\32	v	е	r	у	\0	
									l

这4个字符串常量位于 内存的静态存储区, 只能读取,不能写入。

g

1022

1023

 $\mathbf{a}$ 

1021

 $\mathbf{u}$ 

1020

1019

输出: You

指针变量c[i](1级)

基类型: char型 基类型占空间: 1字节 (实际上不存在,只是一种理解方式)

c[0]	1000	2000
c[1]	1025	2004
c[2]	1035	2008
c[3]	1039	2012

\0

#### cout << (\*--\*++pp+4);

c 2000 2100

指针变量c[i](1级)

基类型: char型

基类型占空间:1字节

(实际上不存在, 只是一

种理解方式)

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

 p[0]
 2012
 2200

 p[1]
 2008
 2204

 p[2]
 2004
 2208

 p[3]
 2000
 2212

指针变量p[i](2级)

基类型:指向char型指针 基类型占空间:4字节

(实际上不存在, 只是一

种理解方式)

p 2200 2300

指针变量p(3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

pp 2204 2400

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间: 4字节

等价: cout << ((\*(\*(pp=pp+1)=\*(pp=pp+1)-1))+4);

c 2000

2100

指针变量c[i](1级)

基类型: char型

基类型占空间: 1字节

(实际上不存在, 只是一

种理解方式

指针变量c(2级)

基类型:指向char型指针

基类型占空间: 4字节

 p[0]
 201
 2
 2200

 p[1]
 200
 8
 2204

 p[2]
 2000
 2208

 p[3]
 2000
 2212

指针变量p[i](2级)

基类型: 指向char型指针

基类型占空间: 4字节

(实际上不存在,只是一

种理解方式)

p 2200 2300

指针变量p (3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

pp 2208 2400

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间:4字节 等价: cout << ((\*(\*(pp=pp+1)=\*(pp=pp+1)-1))+4);

J	o	h	n	\32	1	e	•••	a	n	g
1000	1001	1002	1003	1004	1005	1006	•••	1017	1018	10
В	е	\32	w	е	1	1	!	!	\0	
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	
Y	О	u	\0							
Y 1035	o 1036	u 1037	\ <b>0</b>							ì
			•	v	e	r	у	\0		j

这4个字符串常量位于 内存的静态存储区, 只能读取,不能写入。

11

1020

 $\mathbf{a}$ 

1021

g

1022

1023

输出: \32learn\32C++\32language (为了表述清晰,把"空格"写成\32,下同)

指针变量c[i](1级)

基类型: char型 基类型占空间: 1字节 (实际上不存在,只是一种理解方式)

	100	1
c[0]	1000	2000
c[1]	1025	2004
c[2]	1035	2008
c[3]	1039	2012

\0

#### cout << (\*pp[-2]+3);

c 2000 2100

指针变量c(2级)

基类型:指向char型指针

基类型占空间: 4字节

指针变量c[i](1级)

基类型: char型

基类型占空间:1字节

(实际上不存在, 只是一

种理解方式)

p[0]	2012	2200
p[1]	2008	2204
p[2]	2000	2208
p[3]	2000	2212

p 2200 2300

指针变量p(3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

指针变量p[i](2级) 基类型:指向char型指针 基类型占空间:4字节

(实际上不存在,只是一种理解方式)

pp 2208 2400

指针变量pp (3级) 基类型:指向'指向 char型指针'的指针 基类型占空间: 4字节 

 \$\left(0)\right]\$
 1000
 2000

 \$c[1]\right]\$
 1025
 2004

 \$c[2]\right]\$
 1035
 2008

 \$c[3]\right]\$
 1042
 2012

等价: cout << ((\*(\*(pp-2)))+3);

c 2000 2100

指针变量c[i](1级)

基类型: char型

基类型占空间: 1字节

(实际上不存在,只是-

种理解方式》

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

 p[0]
 2012
 2200

 p[1]
 2008
 2204

 p[2]
 2000
 2208

 p[3]
 2000
 2212

指针变量p[i](2级)

基类型:指向char型指针

基类型占空间: 4字节

(实际上不存在,只是一种理解方式)

p 2200 2300

指针变量p(3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

pp 2200 2400

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间:4字节

#### 等价: cout << ((\*(\*(pp-2)))+3);

J	О	h	n	\32	1	e	•••	a	n
1000	1001	1002	1003	1004	1005	1006	•••	1017	1018
В	е	\32	w	е	1	1	!	!	\0
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034
Y			\ 0						
1	0	u	\0						
1035	1036	1037	1038						
			•	v	e	r	у	\0	

这4个字符串常量位于 内存的静态存储区, 只能读取,不能写入。

g

1022

1023

 $\mathbf{a}$ 

1021

 $\mathbf{u}$ 

1020

1019

输出: very

指针变量c[i](1级)

基类型: char型 基类型占空间: 1字节 (实际上不存在,只是一种理解方式)

c[0]	1000	2000
c[1]	1025	2004
c[2]		2008
c[3]	1000	2 2 112

\0

### cout << (pp[-1][-1]+2);

c 2000 2100

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

指针变量c[i](1级)

基类型: char型

基类型占空间:1字节

(实际上不存在, 只是一

种理解方式)

p[0]	2012	2200
p[1]	2008	2204
p[2]	2000	2208
p[3]	2000	2212

指针变量p[i](2级)

基类型:指向char型指针

基类型占空间: 4字节 (实际上不存在,只是一种理解方式) p 2200 2300

指针变量p (3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

pp 2208 2400

指针变量pp (3级) 基类型:指向'指向 char型指针'的指针 基类型占空间:4字节

等价: cout << ((\*((\*(pp-1))-1)))+2);

c 2000 2100

指针变量c(2级)

基类型: 指向char型指针

基类型占空间: 4字节

指针变量c[i](1级)

基类型: char型

基类型占空间: 1字节

(实际上不存在, 只是一

种理解方式

p[0]	20/2	2200
p[1]	2 200	<b>4</b> _204
p[2]	2000	2208
p[3]	2000	2212

指针变量p[i](2级)

基类型:指向char型指针 基类型占空间:4字节

(实际上不存在,只是一种理解方式)

p 2200 2300

指针变量p (3级)

基类型:指向'指向char型指

针'的指针

基类型占空间: 4字节

pp 2204 2400

指针变量pp(3级) 基类型:指向'指向 char型指针'的指针 基类型占空间:4字节

#### 等价: cout << ((\*((\*(pp-1))-1)))+2);

			_			_		_			
J	0	h	n	\32	1	e	•••	a	n	g	u
1000	1001	1002	1003	1004	1005	1006	•••	1017	1018	1019	102
В	е	\32	w	е	1	1	!	!	\0		
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034		
Y	О	u	\0								
1035	1036	1037	1038							这	
N	0	t	\32	v	е	r	У	\0		内	
1039	1040	1041	1042	1043	1044	1045	1046	1047		只	能

这4个字符串常量位于 内存的静态存储区, 只能读取,不能写入。

 $\mathbf{a}$ 

1021

g

1022

1023

输出: \32well!!

指针变量c[i](1级)

基类型: char型 基类型占空间: 1字节 (实际上不存在,只是一种理解方式)

c[6]	1000	2000
c[1]	1020	2004
c[2]	1035	2008
c[3]	1039	2012

\0

### 综上,输出为:

You learn C++ language very well!!