1. [问答题]

题目描述

给出一个二叉树,用一个函数确定是否有一条从根节点到叶子节点的路径，这个路径上所有节点的值加在一起等于给定的sum的值。函数声明hasPathSum已经给出，写出程序设计思路并且实现该函数。尽量提供多种实现方法。

/\*\*

\* 二叉树节点定义如下：

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

bool hasPathSum(TreeNode \*root, int sum) {

}

示例：

给定的二叉树和 sum = 22,

5

/ \

4 8

/ / \

11 13 4

/ \ \

7 2 1

返回值是ture, 即存在一条 root-to-leaf 的路径 5->4->11->2 节点值相加等于22

2. [问答题]

题目描述

请编写一个函数将字符串s2添加到字符串s1的末端，函数不受s1、s2空间大小的限制。可以利用常用字符串函数strlen,strcpy,strcat,strcmp,strstr实现。

常用字符串函数简单描述:

strlen(char \*str) : 求字符串长度。

strcpy(char \*dest, char \*src) : 把src拷贝到dest。

strcat(char \*dest, char \*src) : 把src连接到dest后面。

strcmp(char \*s1, char \*s2) : 按照各个字符（ascii）比较s1和s2，相等则返回0，否则返回ascii相减的结果。

strstr(char \*s1, char \*s2) : 在s1中查找s2，返回找到的位置，若找不到则返回NULL。

3. [填空题]

下面程序一共会在屏幕上输出多少个“-” ？

#include<iostream>

#include<stdio.h>

#include<sys/types.h>

#include<unistd.h>

using namespace std;

int main( )

{

int i;

for(i = 0; i < 2; i++)

{

cout<<"-\n";

fork( );

cout <<"-\n";

}

cout << endl;

return 1;

}

4. [填空题]

读下面程序，请给出test（ ）函数的返回值。

int test( )

{

int k=0;

char c='A';

do{

switch (c++)

{

case 'A': k++; break;

case 'B': k--;

case 'C': k+=2; break;

case 'D': k=k%2; break;

case 'E': k=k\*10; break;

default: k=k/3;

}

k++;

}while(c<'G');

return k;

}

5. [填空题]

该段代码的输出结果是？

#include<iostream>

using namespace std;

int nest(int i)

{

if (i < 0 )

return 0;

else if (i == 0)

return 1;

else

return nest(i-1) + nest(i-2) + i;

}

int main( )

{

cout << nest(7)<< endl;

return 1;

}

6. [单选题]

如果一个类class A{}没有构造函数定义，那么当这个类的Objects被创建时会发生什么？（ ）

编译器产生错误

运行时会产生错误

编译器会为这个类提供默认构造函数去创建Objects

以上都不对

7. [单选题]

将域名转换为IP地址是由( )服务器完成？

WINS

DHCP

DNS

IIS

8. [单选题]

下面协议中用于WWW传输控制的是？ （ ）

URL

SMTP

HTTP

HTML

9. [单选题]

缺省的Linux系统中，从后台启动进程，应在命令的结尾加上哪个符号？ （ ）

@

&

^

$

10. [单选题]

阅读下面代码：

class B

{

public:

virtual void Fun(){}

};

class D: public B

{

public:

void Fun(){}

};

D dd;

B\* pb = &dd;

D\* pd = &dd;

pb->Fun();

pd->Fun();

上述例程调用的Fun函数顺序为？（ ）

B::Fun, D::Fun

B::Fun, B::Fun

D::Fun, D::Fun

D::Fun, B::Fun

11. [单选题]

如果定义如下类：

class Empty{}

请选择编译器为之生成的函数有哪些？ （ ）

Empty() { … }

Empty(const Empty& rhs){ … }

Empty& operator=(const Empty& rhs) { … }

~Empty() { … }

124

1234

123

14

12. [单选题]

下面程序的输出是？（ ）

int main(void)

{

enum team { my , your = 9 , his , her = his + 3};

printf("%d %d %d %d\n",my , your , his , her);

return 0;

}

0 9 10 13

8 9 11 13

8 9 10 13

0 9 12 9

13. [单选题]

以下表达式选择结果。（ ）

std::string str1("trend");

std::string str2("micro");

std::string& strs = str1;

std::string\* ptrs = &str1;

strs = str2;

ptrs = &str2;

strs = micro, \* ptrs = micro

strs = micro, \* ptrs = trend

strs = trend, \* ptrs = micro

strs = trend, \* ptrs = trend

14. [单选题]

请选择正确的数字填充：某缓存系统采用LRU（近期最少使用算法）淘汰算法，假定缓存容量为4，并且初始为空，那么在顺序访问以下数据项的时候1,5,1,3,2,4,1,2出现缓存命中的次数是\_\_。最后缓存中即将准备淘汰的数据项是（ ）

2, 2

3, 2

3, 3

3, 4

15. [单选题]

请选择表达式 '0' + 1 的结果(本题数值均为十进制)？（ ）

2

31

32

49

16. [单选题]

有关多线程，多进程的描述错误的是？（ ）

子进程获得父进程的数据空间，堆和栈的复制品

线程可以与同进程的其他线程共享数据，但是它拥有自己的栈空间且拥有独立的执行序列

线程执行开销小，但是不利于资源管理和保护

进程是CPU调度和分派的基本单位。

17. [单选题]

分时系统中，当用户数目为100时，为保证响应时间不超过2s，此时的时间片最大应该为？（ ）

10ms

20ms

100ms

200ms

18. [单选题]

设有语句 char a = '\72'; 则关于变量a的说法正确的是？（ ）

包含2个字符

说明不合法

包含1个字符

包含3个字符

19. [单选题]

以下哪项不属于STL container？（ ）

stack

queue

multimap

string

20. [单选题]

以下表达式选择结果。（ ）

int a = 0;

int b = (a=-1) ? 2:3;

int c = (a=0) ? 2:3;

b=2, c=2

b=3, c=3

b=2, c=3

b=3, c=2

21. [单选题]

多个进程怎样共享一个监听端口？（ ）

每个进程都使用SO\_REUSEADDR选项，然后绑定同一个端口

每个进程分别绑定一次即可

使用fork共享

只有最后一个绑定端口的进程才能收到数据

22. [单选题]

从二叉搜索树中查找一个元素时，其时间复杂度大致为（ ）。

O(n)

O(1)

O(log2n)

O(n2)

23. [单选题]

（ ）命令可以查看视图的创建语句 ？

SHOW VIEW

SHOW CREATE VIEW

SELECT VIEW

DISPLAY VIEW

24. [单选题]

对于int \*pa[5]; 的描述，正确的是？（ ）

pa是一个具有5个元素的指针数组，每个元素是一个int类型的指针；

pa[5]表示某个数组的第5个元素的值；

pa是一个指向数组的指针，所指向的数组是5个int类型的元素；

pa是一个指向某个数组中第5个元素的指针，该元素是int类型的变量；

25. [单选题]

关于纯虚函数，下列表述正确的是？（ ）

派生类必须实现基类的纯虚函数

纯虚函数的声明总是以“=0”结束

纯虚函数是给出实现版本（即无函数体定义）的虚函数

含有纯虚函数的类一定是派生类

答案：

2.char\* append(char\*str1,char\*str2)

{

char\* temp;

int length=strlen(str1)+strlen(str2); //计算str1与str2总共的和

temp=new char[length+1]; //申请一段足够容下str1和str2的字符串空间

strcpy(temp,str1); //先将str1复制到字符指针temp指向的内存的起始位置

strcat(temp,str2); //再将str2链接到str1的位置

return temp;

}

3.9

4.8

5.100

6.C

7.C

8.C

9.B

10.C

11.B

12.A

13.A

14.C

15.D

16.D

17.B

18.C

19.D

20.C

21.C

22.C

23.B

24.A

25.B