# 太能沃面试题答案

一、

1.

#include<iostream>

using namespace std;

int main()

{

int32 x = 0;

int32 \*p = &x;

int32 \*q = p+1;

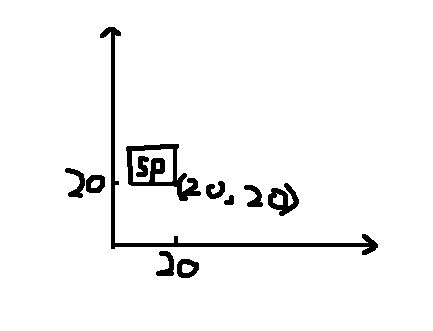
int y = (int)q-(int)p;

cout << y << endl;

}

结果：int y = (int)q-(int)p;这句会有错误，不能将int\* 强转成int；如果不考虑这一点，应该是1.

二、

1. 

2.

当我们在堆上分配一块内存空间的时候，这个对象的引用计数就是1，当有对象要引用这块内存空间的时候，这个引用计数就增加1，当有对象不再引用这块内存的时候引用计数就减1，当这个引用计数减为0的时候就使用delete删除掉这块内存，这样就做到了当有对象引用的时候会正常的访问这块内存，引用完毕也可以正常的回收。

3.

a)直接给node注册一个触摸

b)将node设置为可触摸的，然后设置起触摸形式和回调

4.

cocos2d-x的渲染是先渲染父节点，再将子节点根据层级进行渲染，层级高的最后渲染，然后递归调用。

5.

首先触摸响应会现响应级别较高的，如果级别一样，会根据轴自上而下的传递，如果吞噬，则不会向下进行传递。否则知道全部响应结束。

三、

1.

int func(int a[],int n)

{

int Max\_1 = 0,Max\_2 = 0;

for (int i = 0; i<n; ++i) {

if (a[Max\_1] < a[i]) {

Max\_2 = Max\_1;

Max\_1 = i;

}

}

return Max\_2;

}

2.

题意不清。

3.

local function sort(\_table)

local \_table\_ = {}

local \_pos = 1

table.insert(\_table\_, \_pos,\_table[1])

for i=2,#\_table do

if \_table\_[1].rate < \_table[i].rate then

\_pos = 1

elseif \_table\_[1].rate == \_table[i].rate then

print("==")

if \_table\_[1].star > \_table[i].star then

\_pos = 1

else

for j=2,#\_table\_ do

print("=================")

dump(\_table\_[j])

dump(\_table[i])

print("=================")

if \_table\_[j].rate < \_table[i].rate then

\_pos = j

break

elseif \_table\_[j].rate == \_table[i].rate then

if \_table\_[j].star > \_table[i].star then

\_pos = j

break

end

else

\_pos = #\_table\_ +1

break

end

end

end

else

\_pos = #\_table + 1

end

print(\_pos)

table.insert(\_table\_, \_pos,\_table[i])

end

return \_table\_

end

4.

function split(str,reps)

local resultStrlist={}

string.gsub(str,'[^' .. reps ..']+', function(w)

table.insert(resultStrlist,w)

end)

return resultStrlist;

end

5.

#include <vector>

#include<iostream>

using namespace std;

vector<int>\* func(char\* pstr,vector<int>\* \_v)

{

int length = strlen(pstr);

char\* str = new char[length+1];

strcpy(str, pstr);

int s\_t=0;

char \*tokenPtr=strtok(str,";");

while(tokenPtr!=NULL)

{

s\_t =atoi(tokenPtr);

\_v->push\_back(s\_t);

tokenPtr=strtok(NULL,";");

}

return \_v;

}

int main()

{

char\* str = "1;2;3;4;5";

vector<int> int\_vector;

func(str,&int\_vector);

std::vector<int>::iterator p;

p = int\_vector.begin();

for( ; p!= int\_vector.end(); p++ )

{

cout<<\*p<<endl;

}

return 0;

}

6.

local spawn = cc.Spawn:create(cc.MoveBy:create(1, cc.p(-100,0)),cc.ScaleTo:create(1, 0.25))

local roate = cc.RotateTo:create(1, 30, 0)

local seq = cc.Sequence:create(spawn,roate)

sp:runAction(seq)

7.

local seq = cc.Sequence:create(cc.MoveBy:create(1, cc.p(-100,0)),cc.ScaleTo:create(2, 0.25))

sp:runAction(seq)

8.

local roate = cc.RotateTo:create(1, 0, -180)

sp:runAction(roate)