打算AS

#include <iostream>

#include <string>

using namespace std;

class A

{

public:

virtual std::string WhoAmI(){return "I am A.";}

};

class B : public A

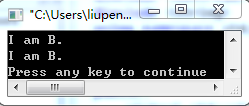
{

public:

virtual std::string WhoAmI(){return "I am B.";}

};

int main()



{

B\* b = new B();

A\* a = (A\*) b;

std::cout << a->WhoAmI() << std::endl;

std::cout << b->WhoAmI() << std::endl;

return 0;

}

#include <iostream>

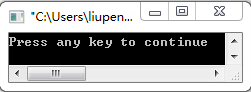
using namespace std;

void GetMemory(char \*p)

{

p = (char\*)malloc(100);

}



int main()

{

char\* str = NULL;

GetMemory(str);

strcpy(str, "hello world");

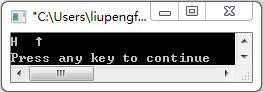
cout << str << endl;

return 0;

}

#include <iostream>

using namespace std;



char \*GetMemory(void)

{

char p[] = "hello world";

return p;

}

int main()

{

char\* str = NULL;

str = GetMemory();

cout << str << endl;

return 0;

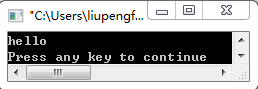
}

#include <iostream>

using namespace std;

void GetMemory(char \*\*p, int num)

{



\*p = (char\*)malloc(num);

}

int main()

{

char\* str = NULL;

GetMemory(&str, 100);

strcpy(str, "hello");

cout << str << endl;

return 0;

}

#include <iostream>

using namespace std;

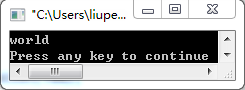
int main()

{

char\* str = (char\*)malloc(100);

strcpy(str, "hello");

free(str);



if(str != NULL)

{

strcpy(str, "world");

cout << str << endl;

}

return 0;

}

#include <iostream>

using namespace std;

struct MyData

{

int A;

int B;

short C;

\_\_int64 D;

};

MyData Func1(MyData data)

{

data.D = data.A + data.B \* data.C;

return data;

}

MyData Func2(MyData& data)

{

data.D = data.A + data.B \* data.C;

return data;

}

int main()

{

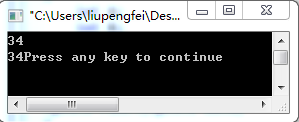
MyData data;

data.A = 10;

data.B = 12;

data.C = 2;

data.D = 100;



data = Func1(data);

// cout << data.D << endl;

printf("%d",data.D);

cout << endl;

data = Func2(data);

// cout << data.D << endl;

printf("%d",data.D);

return 0;

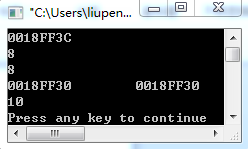
}

#include <iostream>

using namespace std;

int main()

{



int a[2][3] = {0,2,4,6,8,10};

cout << \*(a+1) <<endl;

cout << \*(a[1]+1) <<endl;

cout << \*\*(a+1)+2 <<endl;

cout << a << " " << a[0] <<endl;

cout << a[1][2] << endl;

return 0;

}

//\*优先级大于+

#include <stdio.h>

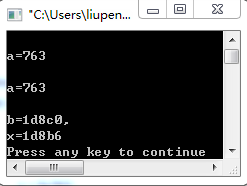
int main()

{

unsigned a,b,x;

int n;

a = 0x763;



n = 6;

b = a << (12-n);

printf("\na=%x\n",a);

x = (a >> n-2)^b;

printf("\na=%x\n",a);

printf("\nb=%x,\nx=%x\n", b,x);

return 0;

}

// 32位系统别忘了

//^按位异或：对应二进位相异时，结果为1

#include <stdio.h>

struct m {

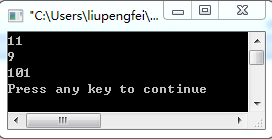
int x;

int \*y;

}\*p;

int a[4] = {12, 33, -40, 100};

struct m b[4] = {10, &a[2], 9, &a[3], 8, &a[0], 7, &a[1]};



int main()

{

p = b;

printf("%d\n", ++p->x);

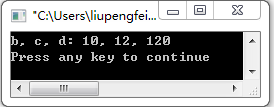
printf("%d\n", (++p)->x);

printf("%d\n", ++(\*p->y));

return 0;

}

#include <iostream>



using namespace std;

int main()

{

int a=10,b,c,d;

b = a++;

c = ++a;

d = 10\*a++;

cout << "b, c, d: " << b << ", " << c << ", " << d << endl;

return 0;

}

#include <iostream>

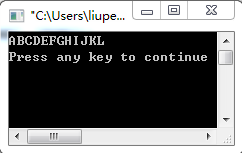
using namespace std;

int main()

{

char \*s[6] = {"ABC","DEF", "GHI", "JKL", "MNO", "PQR"};

char \*\*p;



p=s;

for (int i = 0; i < 4; i++)

cout << p[i];

cout << endl;

return 0;

}

void \*p = malloc(100); sizeof(p) =4