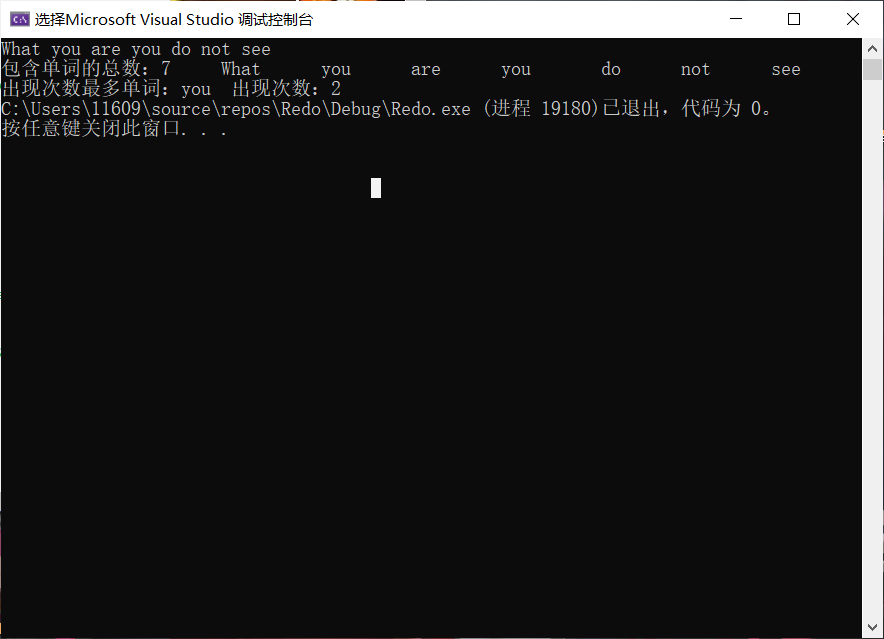
1. **A**



#include<cstdio>

#include<iostream>

using namespace std;

int main()

{

char str[100];

const char\* store[100] = {NULL};

int cmp[100] = {0};

gets\_s(str);

char\* p = str;

bool in = false;

int k = 0;

for (;\*p != 0; p++)

{

if (\*p != ' ')

{

if (!in)

{

in = true;

store[k] = p;

k++;

}

}

else if (\*p == ' ')

{

\*p = 0;

in = false;

}

}

printf("包含单词的总数：%d",k);

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

printf("%9s", store[i]);

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

for (int j = i + 1; j < k; j++)

{

if (strcmp(store[i], store[j]) == 0)

cmp[i]++;

}

int max = 0;

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

{

if (cmp[max] < cmp[i])

max = i;

}

printf("\n出现次数最多单词：%s 出现次数：%d", store[max],cmp[max]+1);

system("pause");

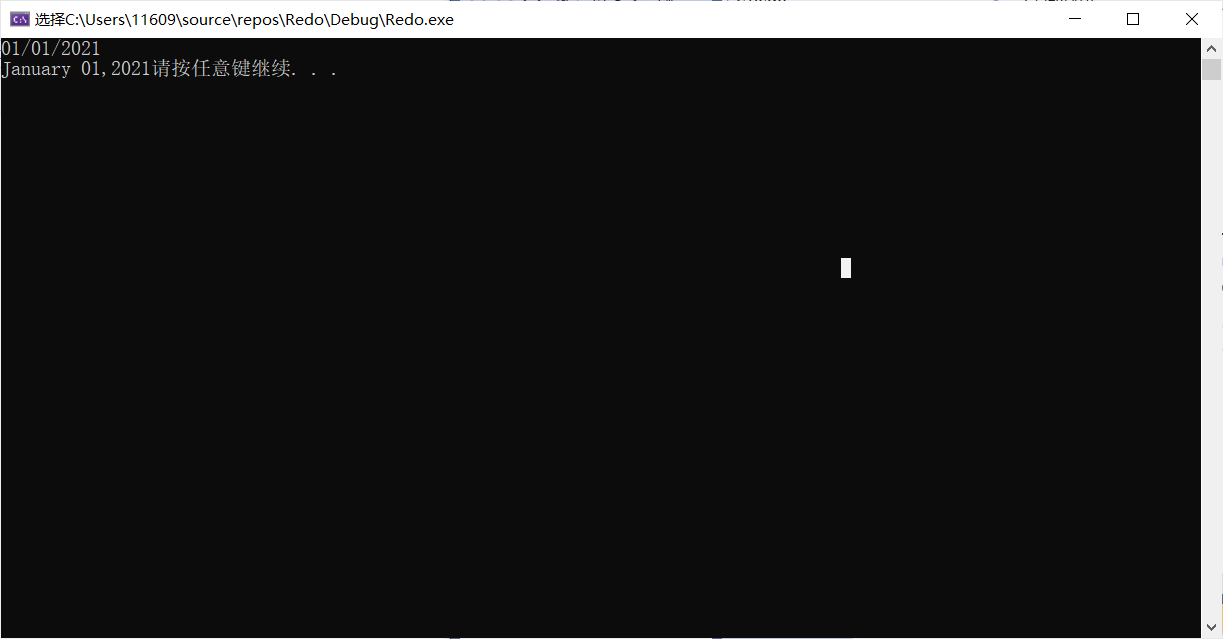
return 0;

}

问题与思考：

想过一些比较诡异的操作，像scanf\_s（vs2019）,printf的返回值实际上就是相应正确输入/输出的参数个数，也许能利用这一点？

**8.B**



#include<cstdio>

#include<iostream>

using namespace std;

int main()

{

char str[20] = {0};

char\* py = &str[6], \* pm = &str[0],\* pd = &str[3];

gets\_s(str);

for (int i = 0; i < 20; i++) if (str[i] == '/') str[i] = 0;

const char\* str2[12] = {"January","February","March","April","May","June","July","August","September","October","November","December" };

printf("%s%3s,%s", str2[atoi(pm)-1],pd,py);

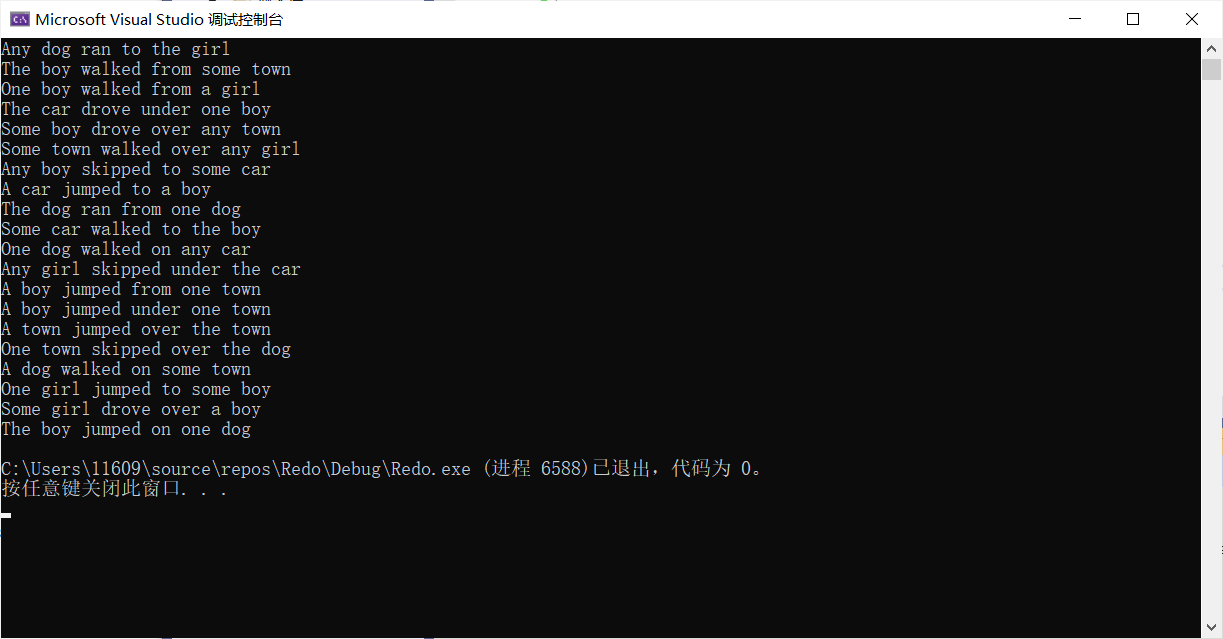
system("pause");

return 0;

}

问题与发现：在vs2019下要求gets必加 \_s,char\*数组前必加const，不然编译会报错，不过不会告诉怎么改；加了\_s可以防止gets一直读取下去，不至于覆盖原先内容，造成破坏；string类型的数组不能在主函数外定义，虽然它的定义上可以采取像定义一般数组一样的形式，但是实际是个类一样的，采用构造函数定义。

**8.C**



#include<iostream>

#include<cstdio>

#include<ctime>

#include<cstdlib>

using namespace std;

int main()

{

const char\* articles[] = { "the","a","one","some","any" };

const char\* noun[] = { "boy","girl","dog","town","car" };

const char\* verbs[] = { "drove","jumped","ran","walked","skipped" };

const char\* prepositions[] = { "to","from","over","under","on" };

int str2[6] = {0};

char support[50] = {0};

srand(time(0));

for (int i = 1; i <= 20; i++)

{

for (int i = 0; i < 6; i++)

str2[i] = rand() % 5;

support[str2[0]] = \*articles[str2[0]] - 32;

printf("%c%s %s %s %s %s %s\n",support[str2[0]],articles[str2[0]]+1, noun[str2[1]], verbs[str2[2]], prepositions[str2[3]], articles[str2[4]], noun[str2[5]]);

}

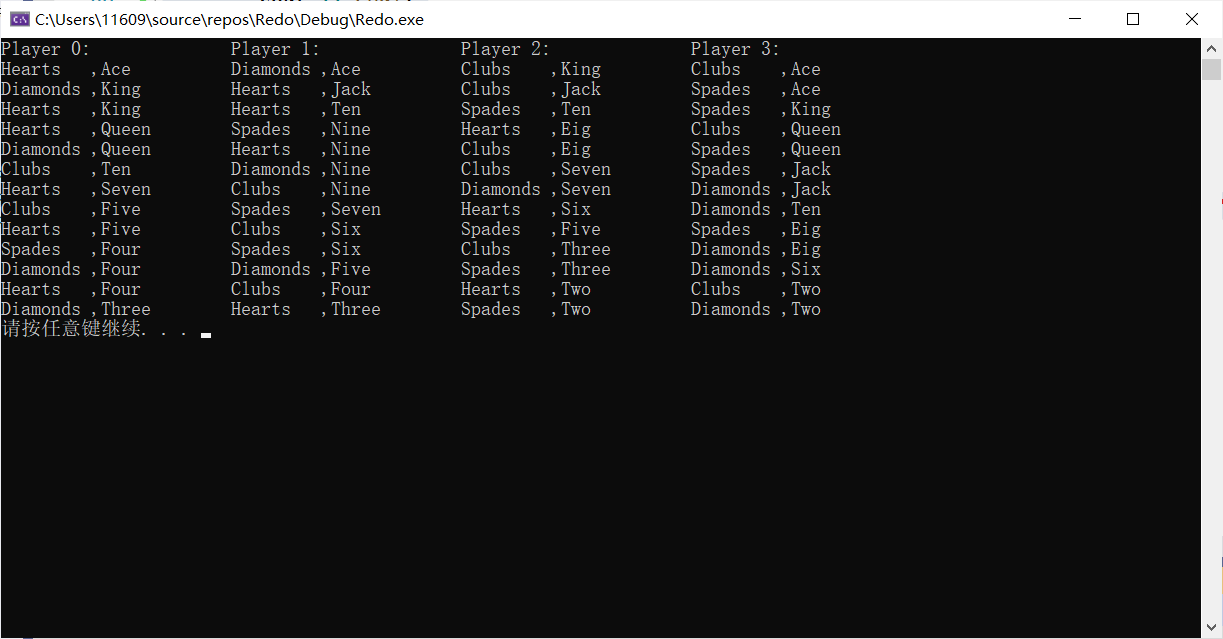
system("pause");

return 0;

}

问题与发现：char\*指针配上printf在字符串的输出操作里显得无比顺手，%后加上数字完全可以取代setw，还可+-对应右左对齐。

**8.D**



#include<iomanip>

#include<iostream>

#include<cstdio>

#include<ctime>

#include<cstdlib>

using namespace std;

class Card

{

public:

int value,suit;

Card(int v = 0,int s = 0)

{

value = v, suit = s;

}

};

class Player

{

public:

Card card[13];

int start;

Player()

{

for (int i = 0; i < 13; i++)

card[i] = Card(0, 0);

start = 0;

}

Player(Card c[], int s = 0)//0,13,26,39

{

int k = 0;

for (int i = s; i < s + 13; i++)

card[k] = c[i], k++;

start = s;

}

};

void shuffle(Card card[]) //考虑random\_shuffle

{

for (int i = 0; i < 52; i++)

swap(card[i], card[(rand() % (52 - i)) + i]);

//1到n的随机交换每张牌概率并不相等，i到n交换可行

}

void Mysort(Card card[])

{

int max = 0;

for (int i = 0; i < 13; i++)

{

max = i;

for (int j = i+1; j < 13; j++)

{

if (card[max].value < card[j].value)

max = j;

}

swap(card[i], card[max]);

}

}

void print(Player player[], const char\* suit[], const char\* value[])

{

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

Mysort(player[i].card);

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

cout << "Player " << i << ": " << setw(20);

cout << endl;

for (int i = 0; i < 13; i++)

{

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

{

printf("%-9s,%-13s", suit[player[k].card[i].suit], value[player[k].card[i].value]);

}

cout << endl;

}

}

int main()

{

srand(time(0));

Card card[52];

const char\* suit[4] = { "Hearts","Diamonds","Clubs","Spades" };

const char\* value[13] = {"Two","Three","Four","Five","Six","Seven","Eig","Nine","Ten","Jack","Queen","King","Ace"};

int k = 0;

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

for(int j = 0;j < 13;j++)

card[k] = Card(j,i),k++;

shuffle(card);

Player player[4];

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

player[i] = Player(card, i \* 13);

print(player, suit, value);

system("pause");

return 0;

}

问题与发现：定义类的数组似乎是不需要空白构造的，只要能提供所有参数的初始值，但是上方Player数组中将Card类数组作为了数据成员，我没法给传进来的指针赋初始值，所以这个空白构造应该是逃不掉。