Codeblocks使用快捷键：

Ctrl+F9 编译

F9 编译+运行

F8 debug

F7 单步调试

scanf()函数返回的值为：正确按指定格式输入变量的个数

1. substr使用

#include <iostream>

#include <stdio.h>

#include<string>

using namespace std;

int main()

{

string s = "123456789";

string sub = s.substr(0,5);

cout<<sub<<endl;

return 0;

}

1. printf只能输出[**C语言**](http://lib.csdn.net/base/c)内置的数据，而string不是内置的，只是一个扩展的类。

#include <stdio.h>

#include<string>

using namespace std;

int main()

{

char a[] = "123456789";

printf("%s",a);

return 0;

}

1. char\* str=(char\*)malloc(15\*sizeof(char)); //动态申请15个char大小的空间

scanf("%s",str);

1. 输入字符串

int main()

{

char sub[100];

scanf("%s",&sub);

cout<<sub<<endl;

return 0;

}

1. Leetcode 回文数判断

#include <iostream>

#include <stdio.h>

#include <string>

using namespace std;

string longestPalindrome(string s)

{

if(s.empty()) return "";

if(s.size()==1) return s;

int min\_start = 0,max\_len = 1;

int i;

for(i=0;i<s.size();)

{

if(s.size()-i <= max\_len/2) break;

int j = i,k = i;

while(k<s.size()-1&&s[k+1]==s[k]) ++k;

i=k+1;

while(k<s.size()-1&&j>0&&s[k+1]==s[j-1]){k++;j--;}

int new\_len = k - j + 1 ;

if(new\_len>max\_len){min\_start = j;max\_len = new\_len;}

}

return s.substr(min\_start,max\_len);

}

int main()

{

char sub[100];

while(scanf("%s",&sub) == 1)

{

string s;

s = longestPalindrome(sub);

cout<<s<<endl;

}

return 0;

}

1. 数字反转

int reverse (int x)

{

long long res = 0;

while(x)

{

res = res \* 10 + x % 10;

x /= 10;

}

return res;

}

1. 输出bool型 printf（”%d”,(x==y)）; true为1，false为0；
2. Leetcode ZigZag形式

class Solution {

public:

string convert(string s, int numRows) {

if(numRows <= 1 )

return s;

const int len = (int)s.length();

string \*str = new string[numRows];

int row = 0,step = 1;

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

str[row].push\_back(s[i]);

if(row == 0)

step = 1;

else if(row == numRows-1 )

step = -1;

row += step;

}

s.clear();

for(int j = 0; j < numRows; j++){

s.append(str[j]);

}

delete[] str;

return s;

}

};

1. 算法导论 树的第一个例子，给出深度，小球编号，问小球应该落在哪里?

#include <cstdio>

#include <cstring>

using namespace std;

const int maxd = 20;

int s[1<<maxd]; //将1左移20位 s的大小为2^20-1

int main()

{

int D,I;

while(scanf("%d%d",&D,&I)==2)

{

memset(s,0,sizeof(s));

int k, n = (1<<D)-1;

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

{

k = 1;

for(;;)

{

s[k] = !s[k]; //数据进行反转

k = s[k]?k\*2:k\*2+1; //树的特殊情况

if(k>n) break;

}

}

printf("%d\n",k/2);

}

return 0;

}

1. Leetcode 13

#include <cstdio>

#include <string>

#include <iostream>

using namespace std;

string intToRoman(int num);

int main(){

int num;

while(scanf("%i",&num) ==1){

string str = intToRoman(num);

cout<<str<<endl;

}

return 0;

}

string intToRoman(int num){

string M[] = {"","M","MM","MMM"};

string C[] = {"","C","CC","CCC","CD","D","DC","DCC","DCCC","CM"};

string X[] = {"","X","XX","XXX","XL","L","LX","LXX","LXXX","XC"};

string I[] = {"","I","II","III","IV","V","VI","VII","VIII","IX"};

return M[num/1000]+C[num%1000/100]+X[num%100/10]+I[num%10];

}

11.指针的第一次使用

void swap(int\* a,int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

1. 输出三角形

#include <stdio.h>

int main()

{

int len;

while(scanf("%d",&len) == 1)

{

/\*

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

{

for(int j =0; j<len-i; j++)

{

printf(" ");

}

for(int j=0; j<2\*i-1; j++)

{

printf("\*");

}

printf("\n");

}

\*/

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

{

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

printf(" ");

for(int j=0; j<2\*(len-i)-1; j++)

printf("\*");

printf("\n");

}

}

return 0;

}

1. 循环输入的一个例子

#include <string>

const int maxn = 100 + 5;

char pic[maxn][maxn];

int main()

{

int m,n;

while(scanf("%d %d",&m,&n) == 2 && m && n){

for(int i=0; i<m; i++)scanf("%s",pic[i]);

printf("scanf end\n");

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

{

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

printf("%c",pic[i][j]);

printf("\n");

}

}

return 0;

}