How to check the type of identifiers in C?

[intro]

There are a class in the library <typeinfo> can achieve it.

typeid().name()

[namespace]

NONE

[library]

<typeinfo>

[code]

#include <iostream>

#include <string>

#include <typeinfo>

using namespace std;

struct myStruct

{

int x;

int y;

};

class myClass

{

public:

int x;

int y;

};

int main()

{

int i;

int \* pi;

long int li;

long int \*pli;

long long int lli;

long long int \* plli;

unsigned int ui;

unsigned int \* pui;

float f;

float \* pf;

double d;

double \* pd;

long double ld;

long double \* pld;

char c;

char \* pc;

unsigned char uc;

unsigned char \* puc;

string s="124";

string \* ps;

struct myStruct mS;

struct myStruct \* pmS;

myClass mC;

myClass \* pmC;

cout << "int is: " << typeid(int).name() << endl;

cout << " i is: " << typeid(i).name() << endl;

cout << " pi is: " << typeid(pi).name() << endl;

cout << "\*pi is: " << typeid(\*pi).name() << endl;

cout << "long int is: " << typeid(long int).name() << endl;

cout << " li is: " << typeid(li).name() << endl;

cout << " pi is: " << typeid(pli).name() << endl;

cout << " \*pli is: " << typeid(\*pli).name() << endl;

cout << "long long int is: " << typeid(long long int).name() << endl;

cout << " lli is: " << typeid(lli).name() << endl;

cout << " plli is: " << typeid(plli).name() << endl;

cout << " \*plli is: " << typeid(\*plli).name() << endl;

cout << "unsigned int is: " << typeid(unsigned int).name() << endl;

cout << " ui is: " << typeid(ui).name() << endl;

cout << " pui is: " << typeid(pui).name() << endl;

cout << " \*pui is: " << typeid(\*pui).name() << endl;

cout << "float is: " << typeid(float).name() << endl;

cout << " f is: " << typeid(f).name() << endl;

cout << " pf is: " << typeid(pf).name() << endl;

cout << " \*pf is: " << typeid(\*pf).name() << endl;

cout << "double is: " << typeid(double).name() << endl;

cout << " d is: " << typeid(d).name() << endl;

cout << " pd is: " << typeid(pd).name() << endl;

cout << " \*pd is: " << typeid(\*pd).name() << endl;

cout << "long double is: " << typeid(long double).name() << endl;

cout << " ld is: " << typeid(ld).name() << endl;

cout << " pld is: " << typeid(pld).name() << endl;

cout << " \*pld is: " << typeid(\*pld).name() << endl;

cout << "char is: " << typeid(char).name() << endl;

cout << " c is: " << typeid(c).name() << endl;

cout << " pc is: " << typeid(pc).name() << endl;

cout << " \*pc is: " << typeid(\*pc).name() << endl;

cout << "unsigned char is: " << typeid(unsigned char).name() << endl;

cout << " uc is: " << typeid(uc).name() << endl;

cout << " puc is: " << typeid(puc).name() << endl;

cout << " \*puc is: " << typeid(\*puc).name() << endl;

cout << "string is: " << typeid(string).name() << endl;

cout << " s is: " << typeid(s).name() << endl;

cout << " ps is: " << typeid(ps).name() << endl;

cout << " \*ps is: " << typeid(\*ps).name() << endl;

cout << "struct myStruct is: " << typeid(struct myStruct).name() << endl;

cout << " mS is: " << typeid(mS).name() << endl;

cout << " pmS is: " << typeid(pmS).name() << endl;

cout << " \*pmS is: " << typeid(\*pmS).name() << endl;

cout << "myClass is: " << typeid(myClass).name() << endl;

cout << " mC is: " << typeid(mC).name() << endl;

cout << " pmC is: " << typeid(pmC).name() << endl;

cout << " \*pmC is: " << typeid(\*pmC).name() << endl;

cout<<"------"<<endl;

int iA[2];

long int liA[2];

long long int lliA[2];

float fA[2];

double dA[2];

long double ldA[2];

char cA[2];

unsigned char ucA[2];

string sA[2];

struct myStruct mSA[2];

myClass mCA[2];

cout << " iA is: " << typeid(iA).name() << endl;

cout << " liA is: " << typeid(liA).name() << endl;

cout << " lliA is: " << typeid(lliA).name() << endl;

cout << " fA is: " << typeid(fA).name() << endl;

cout << " dA is: " << typeid(dA).name() << endl;

cout << " ldA is: " << typeid(ldA).name() << endl;

cout << " cA is: " << typeid(cA).name() << endl;

cout << " ucA is: " << typeid(ucA).name() << endl;

cout << " sA is: " << typeid(sA).name() << endl;

cout << " mSA is: " << typeid(mSA).name() << endl;

cout << " mCA is: " << typeid(mCA).name() << endl;

cout<<"-------"<<endl;

return 0;

}

[result]

int is: i

i is: i

pi is: Pi

\*pi is: i

long int is: l

li is:

l

pi is: Pl

\*pli is: l

long long int is: x

lli is: x

plli is: Px

\*plli is: x

unsigned int is: j

ui is: j

pui is: Pj

\*pui is: j

float is: f

f is: f

pf is: Pf

\*pf is: f

double is: d

d is: d

pd is: Pd

\*pd is: d

long double is: e

ld is: e

pld is: Pe

\*pld is: e

char is: c

c is: c

pc is: Pc

\*pc is: c

unsigned char is: h

uc is: h

puc is: Ph

\*puc is: h

string is: NSt3\_\_212basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEE

s is: NSt3\_\_212basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEE

ps is: PNSt3\_\_212basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEE

\*ps is: NSt3\_\_212basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEE

struct myStruct is: 8myStruct

mS is: 8myStruct

pmS is: P8myStruct

\*pmS is: 8myStruct

myClass is: 7myClass

mC is: 7myClass

pmC is: P7myClass

\*pmC is: 7myClass

------

iA is: A2\_i

liA is: A2\_l

lliA is: A2\_x

fA is: A2\_f

dA is: A2\_d

ldA is: A2\_e

cA is: A2\_c

ucA is: A2\_h

sA is: A2\_NSt3\_\_212basic\_stringIcNS\_11char\_traitsIcEENS\_9allocatorIcEEEE

mSA is: A2\_8myStruct

mCA is: A2\_7myClass

-------

[ref]

<https://cplusplus.com/reference/typeinfo/type_info/name/>