

1 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    vector<int> myvector(5);
```

```
    int*p = myvector.data();
```

```
    *p = 10;
```

```
    ++p;
```

```
    *p = 20;
```

```
    p[2] = 100;
```

```
    for (unsigned i = 0; i < myvector.size(); ++i)
```

```
        cout << ' ' << myvector[i];
```

```
    return 0;
```

```
}
```

vì con trỏ p  
đang trỏ tới  
phần tử thứ 2  
của vector nên  
khi truy cập  
tới p[2] thì là  
truy cập tới  
phần tử thứ 3  
trong vector

A. Another Answer

B. 10 20 0

C. 10 20 0 100

D. 10 20

E. 10 20 0 100 0

2 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int var_x = 5;
    try
    {
        if(var_x < 0)
            throw "Error!";
        cout << var_x;
    }
    catch (const char* e)
    {
        cout << e;
    }
    return 0;
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

**D. 5**

E. 15

F. 10

4 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    public:
```

```
        virtual void Func();
```

```
};
```

```
void Base::Func()
```

```
{
```

```
    cout << "1";
```

```
}
```

```
class DerivedA : public Base
```

```
{
```

```
    public:
```

```
        void Func();
```

```
};
```

```
void DerivedA::Func()
```

```
{
```

```
    cout << "2";
```

```
}
```

```
class DerivedB : public DerivedA
```

```
{
```

```
    public:
```

```
        void Func();
```

```
};
```

```
void DerivedB::Func()
```

```
{
```

```
    cout << "3";
```

```
}
```

```
int main()
```

```
{
```

```
    Base* ptr ;
```

```
    Base obj;
```

```
    DerivedA obj_a;
```

```
    DerivedB obj_b;
```

```
    ptr = &obj;
```

```
    ptr->Func();
```

```
    ptr = &obj_a;
```

```
    ptr->Func();
```

```
    ptr = &obj_b;
```

```
    ptr->Func();
```

```
    return 0;
```

```
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error

**D. 123**

6 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    int prop;
```

```
    public:
```

```
        Base();
```

```
        void Func();
```

```
};
```

```
Base::Base()
```

```
    : prop(0)
```

```
{
```

```
}
```

```
void Base::Func()
```

```
{
```

```
    this->prop++;
```

```
    cout << this->prop;
```

```
}
```

```
class Derived : public Base
```

```
{
```

```
};
```

```
int main()
```

```
{
```

```
    Derived obj;
```

```
    obj.Func();
```

```
    return 0;
```

```
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error

**D. 1**

E. 0

7 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    public:
```

```
        virtual void Func();
```

```
};
```

```
void Base::Func()
```

```
{
```

```
    cout << "Base";
```

```
}
```

```
class Derived : public Base
```

```
{
```

```
    private:
```

```
        void Func();
```

```
};
```

```
void Derived::Func()
```

```
{
```

```
    cout << "Derived";
```

```
}
```

```
int main()
```

```
{
```

```
    Base* ptr = new Base();
```

```
    Derived obj;
```

```
    ptr = &obj;
```

```
    ptr->Func();
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

**D. Derived**

E. Garbage Value

8 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
class Base
```

```
{
```

```
    public:
```

```
        Base();
```

```
        ~Base();
```

```
};
```

```
Base::Base()
```

```
{
```

```
    cout << "1";
```

```
}
```

```
Base::~~Base()
```

```
{
```

```
    cout << "2";
```

```
}
```

```
class Derived : public Base
```

```
{
```

```
    public:
```

```
        Derived();
```

```
        ~Derived();
```

```
};
```

```
Derived::Derived()
```

```
{
```

```
    cout << "3";
```

```
}
```

```
Derived::~~Derived()
```

```
{
```

```
    cout << "4";
```

```
}
```

```
int main()
```

```
{
```

```
    Derived obj;
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

D. 4321

**E. 1342**

9 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    try
```

```
    {
```

```
        throw 20;
```

```
    }
```

```
    catch (int e)
```

```
    {
```

```
        cout << "An exception occurred " << e;
```

```
    }
```

```
    return 0;
```

```
}
```

Ném ngo i l

B t ngo i l  
và in ra ngo i  
l

A. Another Answer

B. Runtime Error

C. Compile Error

**D. An exception occurred 20**

E. 20

F. An exception occurred

10 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    double var_x = 10, var_y = 5, Res;
    char ope;
    try
    {
        ope chỉ a
        c gán giá
        tr
        if(ope != '+' && ope != '-' && ope != '*' && ope != '/')
            throw ope;
        switch (ope)
        {
            case '+':
                Res = var_x + var_y;
                break;
            case '-':
                Res = var_x - var_y;
                break;
            case '*':
                Res = var_x * var_y;
                break;
            case '/':
                Res = var_x / var_y;
                break;
        }
        cout << Res;
    }
    catch (const char e)
    {
        cout << "Error!";
    }
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. Error!**
- E. 2



11 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
#include <exception>
using namespace std;
class MyException : public exception
{
    virtual const char* what() const throw()
    {
        return "Error!";
    }
} me;
int main()
{
    try
    {
        throw me;
    }
    catch (exception & e)
    {
        cout << e.what();
    }
    return 0;
}
```

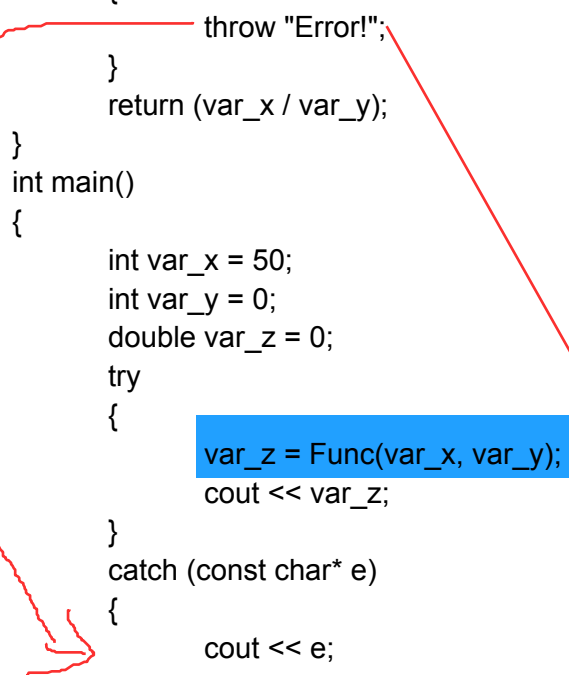
Ném i  
t ng me

nh n i t ng  
me và in ra  
b ng ph ng  
th c

- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. Error!**
- E. exception

12 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
using namespace std;
double Func(int var_x, int var_y)
{
    if(var_y == 0)
    {
        throw "Error!";
    }
    return (var_x / var_y);
}
int main()
{
    int var_x = 50;
    int var_y = 0;
    double var_z = 0;
    try
    {
        var_z = Func(var_x, var_y);
        cout << var_z;
    }
    catch (const char* e)
    {
        cout << e;
    }
    return 0;
}
```



- A. Another Answer
- B. Runtime Error
- C. Compile Error
- D. Error!**
- E. 0
- F. 50

13 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
using namespace std;

class BaseA
{
    private:
        int x;
    public:
        BaseA();
        void Show();
};
BaseA::BaseA()
{
    this->x = 5;
}
void BaseA::Show()
{
    cout << this->x;
}
class BaseB
{
    protected:
        int x;
    public:
        BaseB();
        void Show();
};
BaseB::BaseB()
{
    this->x = 10;
}
void BaseB::Show()
{
    cout << this->x;
}
class Derive : public BaseA, public BaseB
{
};

int main()
{
    Derive obj;
    obj.BaseA::Show();
    system("pause");
    return 0;
}
```

- A. Another Answer
- B. Runtime Error
- C. Compile Error

**D. 5**

14 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
template <typename T, int count>
```

```
void Func(T var_x)
```

```
{
```

```
    T arr[count];
```

```
    for(int i = 0; i < count; ++i)
```

```
    {
```

```
        arr[i] = var_x++;
```

```
        cout << arr[i];
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    double var_x = 2.1;
```

```
    Func<double, 3>(var_x);
```

```
    return 0;
```

```
}
```

Gán giá trị  
trước khi  
in

- A. Another Answer
- B. Runtime Error
- C. Compile Error

D. 3.1

E. 4.1

**F. 2.13.14.1**

15 Kết quả thực hiện đoạn chương trình sau là gì?

```
#include <iostream>
```

```
using namespace std;
```

```
struct BaseA
```

```
{
```

```
    int prop_a;
```

```
};
```

```
struct BaseB
```

```
{
```

```
    int* prop_b;
```

```
};
```

```
struct BaseC : public BaseA, public BaseB
```

```
{
```

```
};
```

```
int main()
```

```
{
```

```
    BaseC* ptr = new BaseC;
```

```
    ptr->prop_b = 0;
```

```
    cout << "Inherited";
```

```
    return 0;
```

```
}
```

A. Another Answer

B. Runtime Error

C. Compile Error

**D. Inherited**