

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>

using namespace std;

int main()
{
    //[문항1]

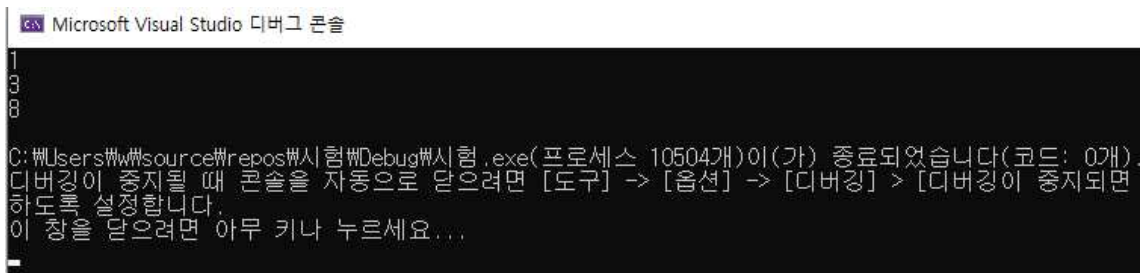
    vector<double> Num = { 1,-2,3,-5,8,-3 };
    vector<double> filter;

    copy_if(Num.begin(), Num.end(), back_inserter(filter), [](const auto& num)
{return num > 0; });

    for (auto i = filter.begin(); i < filter.end(); i++) {
        cout << *i << endl;
    }

    return 0;
}

```



Microsoft Visual Studio 디버그 콘솔

```

1
3
8
C:\Users\www\source\repos\시험\Debug\시험.exe(프로세스 10504개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>
using namespace std;

int main()
{

    //[문항2]

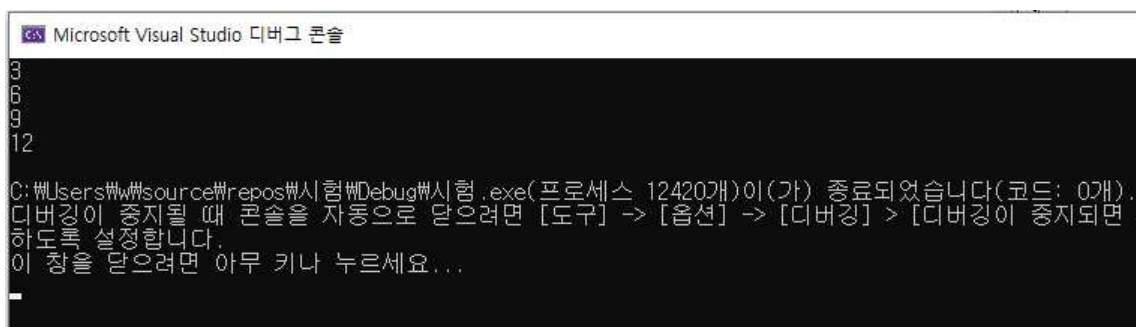
    vector<double> Num = { 1,2,3,4 };
    vector<double> map;

    transform(Num.begin(), Num.end(), back_inserter(map), [](const auto& num)
{return num * 3; });

    for (auto i = map.begin(); i < map.end(); i++) {
        cout << *i << endl;
    }

    return 0;
}

```



```

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>
using namespace std;

int main()
{

    ////[문항3]

    string password, password1;
    int count = 1;
    cout << "암호를 설정하세요 : ";
    cin >> password;
    while (true) {
        cout << "설정한 암호를 입력하세요 : ";
        cin >> password1;
        //strcmp
        if (password.compare(password1) == 0) {
            cout << "정상 종료합니다" << endl;
            break;
        }
        else
        {
            cout << "암호가 틀렸습니다" << endl;
            count++;
        }
        if (count == 6) {
            cout << "암호가 5회 틀렸습니다. 종료합니다!!!" << endl;
            break;
        }
    }

    return 0;
}

```

Microsoft Visual Studio 디버그 콘솔

```
암호를 설정하세요 : abcd
암호를 입력하세요 : ab
암호가 틀렸습니다
암호를 입력하세요 : ad
암호가 틀렸습니다
암호를 입력하세요 : ac
암호가 틀렸습니다
암호를 입력하세요 : ag
암호가 틀렸습니다
암호를 입력하세요 : ae
암호가 틀렸습니다
암호가 5회 틀렸습니다. 종료합니다!!!
```

C:\Users\source\repos\Debug\Debug.exe(프로세스 4816개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면
하도록 설정합니다].
이 창을 닫으려면 아무 키나 누르세요...

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>

using namespace std;

int main()
{
    //[문항4]

    srand((unsigned)time(NULL));

    int number = rand() % 101 + 1; //1부터 100까지
    int user;
    while (true) {
        cout << "숫자를 입력하세요 : ";
        cin >> user;

        if (user > number) {
            cout << "down" << endl;
        }
        else if (user < number) {
            cout << "up" << endl;
        }
        else if (user == number) {
            cout << "정답" << endl;
            break;
        }
    }

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

숫자를 입력하세요 : 40

up

숫자를 입력하세요 : 60

up

숫자를 입력하세요 : 80

down

숫자를 입력하세요 : 70

down

숫자를 입력하세요 : 65

up

숫자를 입력하세요 : 67

정답

C:\Users\www\source\repos\www\Debug\www.exe(프로세스 10980개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 콘솔을 닫으려면] 하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>

using namespace std;

int main()
{
    //[문항5]
    cout << "-----" << endl;
    cout << "숫자 야구 게임" << endl;
    cout << "-----" << endl;
    cout << "컴퓨터가 수비수가 되어 세 자리 수를 하나 골랐습니다." << endl;
    cout << "각 숫자는 0~9 중 하나며 중복되는 숫자는 없습니다"<<endl;
    cout << "모든 숫자와 위치를 맞추면 승리합니다" << endl;
    cout << "숫자와 순서가 둘다 맞으면 스트라이크 입니다" << endl;
    cout << "숫자만 맞고 순서가 틀리면 볼입니다" << endl;
    cout << "숫자가 틀리면 아웃입니다" << endl;
    cout << "-----" << endl;

    srand((unsigned)time(NULL));
    int number[3];
    int user[3];

    //중복없는 랜덤 0~9만듬
    for (int i = 0; i < 3; i++) {
        number[i] = rand() % 10;
        for (int j = 0; j < i; j++) {
            if (number[i] == number[j]) {
                i--;
                break;
            }
        }
    }

    while (true) {
        int ball = 0, strike = 0, out = 3;

```

```

cout << "첫 번째 숫자를 입력하세요 : ";
cin >> user[0];
cout << "두 번째 숫자를 입력하세요 : ";
cin >> user[1];
cout << "세 번째 숫자를 입력하세요 : ";
cin >> user[2];

//같은숫자검사
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < i; j++) {
        if (user[i] == user[j]) {
            cout << "같은 숫자를 입력하면 안됩니다." <<
endl;

            break;
        }
    }
}

//판정
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        if (number[i] == user[j]&& i!=j) {
            strike++; out--;
        }
        else if(number[i] == user[j]){
            ball++; out--;
        }
    }
}

cout << "공격 : " << user[0] << " " << user[1] << " " << user[2] << "
";

cout << "판정 : " << ball << "B" << " " << strike << "S" << " " << out
<< "O" << " " << endl;

//테스트를 위해 정답을 출력함
cout << "정답" << endl;
for (int i = 0; i < 3; i++) {
    cout << number[i] << " ";
}
cout << endl;

```



```

        if (strike == 3) {
            cout << "정답" << endl;
            for (int i = 0; i < 3; i++) {
                cout << number[i] << " ";
            }
            cout << endl;
            break;
        }
    }

    return 0;
}

```

Microsoft Visual Studio 디버그 콘솔

```

숫자 야구 게임
-----
컴퓨터가 수비수가 되어 세 자리 수를 하나 골랐습니다.
각 숫자는 0~9 중 하나며 중복되는 숫자는 없습니다.
모든 숫자와 위치를 맞추면 승리합니다.
숫자와 순서가 둘다 맞으면 스트라이크 입니다.
숫자만 맞고 순서가 틀리면 볼입니다.
숫자가 틀리면 아웃입니다.
-----
첫 번째 숫자를 입력하세요 : 1
두 번째 숫자를 입력하세요 : 3
세 번째 숫자를 입력하세요 : 5
공격 : 1 3 5 판정 : 2B 0S 10
정답
3 6 1
첫 번째 숫자를 입력하세요 : 5
두 번째 숫자를 입력하세요 : 3
세 번째 숫자를 입력하세요 : 1
공격 : 5 3 1 판정 : 1B 1S 10
정답
3 6 1
첫 번째 숫자를 입력하세요 : 3
두 번째 숫자를 입력하세요 : 6
세 번째 숫자를 입력하세요 : 1
공격 : 3 6 1 판정 : 0B 3S 00
정답
3 6 1
정답
3 6 1
C:\Users\www\source\repos\Debug\Debug.exe(프로세스 14164개)이(가) 종료되었습니다(코드: 0개)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <time.h>

using namespace std;
//[문항6]
class Employee {
private:
    string name;

public:
    Employee(string name) {
        this->name = name;
    }
    void ShowYourName() const {
        cout << "이름 : " << name << endl;
    }
    virtual int GetPay() const {
        return 0;
    }
    virtual void ShowSalaryInfo() const {}
};

```

```

class PermanentWorker : public Employee
{
private:
    int salary;
public:
    PermanentWorker(string name, int salary) : Employee(name)
    {
        this->salary = salary;
    }
    int GetPay() const {
        return this->salary;
    }
    void ShowSalaryInfo() const
    {
        ShowYourName();
    }
};

```

```

        cout << "이달의 월급 : " << GetPay() << endl;;
    }
};

```

```

class SalesWorker : public PermanentWorker
{
private:
    int salesResult;
    double bonusRatio;
public:
    SalesWorker(string name, int salary, int bonusRatio) :
    PermanentWorker(name, salary)
    {
        this->salesResult = 0;
        this->bonusRatio = bonusRatio;

    }
    int GetPay() const
    {
        return PermanentWorker::GetPay() + int(salesResult * bonusRatio);
    }
    void ShowSalaryInfo() const
    {
        ShowYourName();
        cout << "이달의 월급(상여금) : " << GetPay() << endl;;
    }

};

```

```

class TemporaryWoker :public Employee
{
private:
    int workTime;
    int payPerHour;

public:
    TemporaryWoker(string name, int payPerHour) : Employee(name),
        workTime(0),

```

```

        payPerHour(payPerHour) {}

void AddWorkTime(int time)
{
    this->workTime += time;
}

int GetPay() const
{
    return payPerHour * workTime;
}

void ShowSalaryInfo() const
{
    ShowYourName();
    cout << "이번달 월급(시급) : " << GetPay() << endl;
}

};

```

```

class EmployHandler {
private:
    Employee* empList[50];
    int empNum;
public:
    EmployHandler() : empNum(0), empList{ 0 } {}
    void AddEmployee(Employee* emp)
    {
        empList[empNum++] = emp;
    }
    void ShowAllSalaryInfo() const
    {
        for (int i = 0; i < empNum; i++) {
            empList[i]->ShowSalaryInfo();
        }
    }
}

```

```

void ShowTotalSalary() const
{
    double sum = 0;
    for (int i = 0; i < empNum; i++) {
        sum += empList[i]->GetPay();
    }
    cout << "직원 급여 총합 : " << sum << endl;

}

~EmployHandler()
{
    for (int i = 0; i < empNum; i++) {
        delete empList[i];
    }
}

};

int main()
{
    //[문항6] main
    EmployHandler hand;

    //직원등록
    hand.AddEmployee(new PermanentWorker("직원1", 200));
    hand.AddEmployee(new PermanentWorker("직원2", 400));
    hand.AddEmployee(new PermanentWorker("직원3", 500));

    TemporaryWoker* parttime = new TemporaryWoker("알바생", 8);
    parttime->AddWorkTime(80);
    hand.AddEmployee(parttime);

    //이번달에 지급해야할 급여의 정보
    hand.ShowAllSalaryInfo();

    //이번달에 지급해야할 급여의 총합
    hand.ShowTotalSalary();

    return 0;
}

```

Microsoft Visual Studio 디버그 콘솔

```
이 : 직원1  
이 : 직원2 : 200  
이 : 직원3 : 400  
이 : 직원4 : 500  
이 : 발생  
이 : 급여(시급) : 640  
직원 급여 총합 : 1740  
  
C:\Users\source\repos\Debug\Debug.exe(프로세스 5648개)이(가) 종료되었습니다(코드: 0개).  
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지될 때 콘솔을 닫음] 하도록 설정합니다.  
이 창을 닫으려면 아무 키나 누르세요...
```