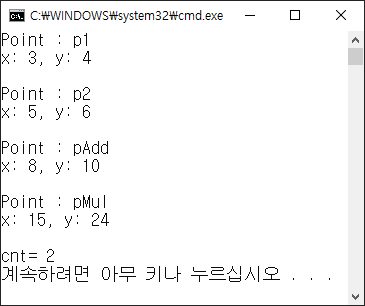
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_C++ \_09 | 학번 : | 20185309 | 이름 : | 황명원 |

* **프로그램 과제**

1. PointCom클래스 멤버 함수와 전역함수 showPoint()를 프렌드 함수로 갖는 Point 클래스를 구현하고 실행하는 프로그램을 완성 하시오

class Point;

void showPoint(const Point &p, const string pname);

class PointCom {

int cnt;

public:

PointCom():cnt(0){}

Point pointAdd(const Point& p1, const Point& p2);

Point pointMul(const Point& p1, const Point& p2);

~PointCom() { cout << "cnt= " << cnt << endl; }

};

class Point {

int x, y;

public:

Point(const int &xp, const int &yp):x(xp), y(yp){}

//이곳에 프렌드 함수 선언

};

int main() {

Point p1(3, 4);

Point p2(5, 6);

PointCom pc;

showPoint(p1, "p1");

showPoint(p2, "p2");

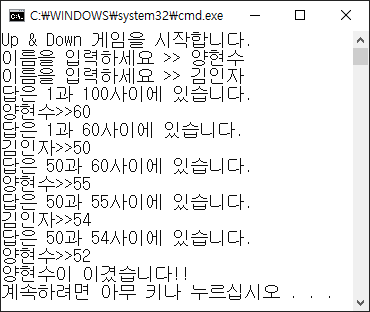
showPoint(pc.pointAdd(p1, p2), "pAdd");

showPoint(pc.pointMul(p1, p2), "pMul");

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Point; void showPoint(const Point &p, const string pname);  class PointCom {  int cnt; public:  PointCom() : cnt(0) {}  Point pointAdd(const Point &p1, const Point &p2);  Point pointMul(const Point &p1, const Point &p2);  ~PointCom() { cout << "cnt= " << cnt << endl; } };  class Point {  int x, y; public:  Point(const int &xp, const int &yp) : x(xp), y(yp) {}  ~Point(){}  //이곳에 프렌드 함수 선언  friend void showPoint(const Point &p, const string pname);  friend Point PointCom::pointAdd(const Point &p1, const Point &p2);  friend Point PointCom::pointMul(const Point &p1, const Point &p2); //이곳에 프렌드 함수 선언 };  Point PointCom::pointAdd(const Point &p1, const Point &p2){  int x;  int y;  x=p1.x+p2.x;  y=p1.y+p2.y;  Point tmp(x,y);  cnt++;  return tmp; } Point PointCom::pointMul(const Point &p1, const Point &p2){  int x;  int y;  x=p1.x\*p2.x;  y=p1.y\*p2.y;  Point tmp(x,y);  cnt++;  return tmp; }  void showPoint(const Point &p, const string pname){  cout<<"Point : "<<pname<<endl;  cout<<"x: "<<p.x<<" y: "<<p.y<<endl; }  int main() {  Point p1(3, 4);  Point p2(5, 6);  PointCom pc;  showPoint(p1, "p1");  showPoint(p2, "p2");  showPoint(pc.pointAdd(p1, p2), "pAdd");  showPoint(pc.pointMul(p1, p2), "pMul"); } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 2명의 사람이 번갈아 숨겨진 답을 찾는 사람이 이긴다. 찾아야 할 답은 난수로 생성하며 static 멤버로 구성하는 UpAndDownGame 클래스와 선수를 다루기 위한 Person 클래스를 구현하여 처리하도록 한다.

class Person {

string name;

public:

Person(string name) { this->name = name; }

string getName() { return name; }

bool go(); // 정수를 입력 받고 정답이면 승리, true 리턴

};

class UpAndDownGame {

static int answer; //맞춰야 할 답, 난수로 초기화

static int top; //맞춰야 할 답의 범위에서 최고값

static int bottom; //맞춰야 할 답의 범위에서 최저값

static void init(); // 난수를 생성하여 answer 변수로 초기화

static void msg(); // top과 bottom 출력

public:

static void run(); // 게임 진행

static bool check(int answer); //top과 bottom을 조절하고, 정답을 맞추었으면 true 리턴

};

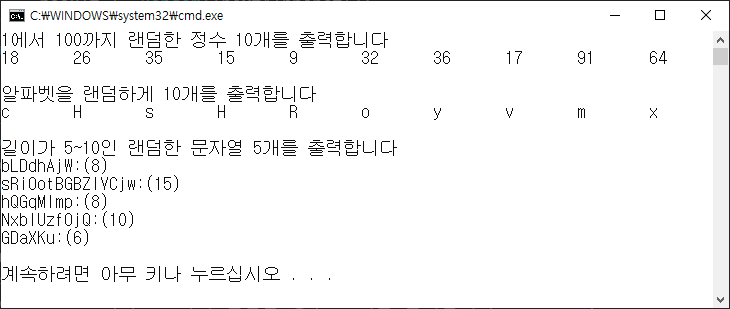
int main() {

UpAndDownGame::run();

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <cstdlib> #include <ctime> #include <string> using namespace std;  class UpAndDownGame;  class Person {  string name; public:  Person(string name) { this->name = name; }   string getName() { return name; }   bool go();// 정수를 입력 받고 정답이면 승리, true 리턴  };  class UpAndDownGame {  static int answer; //맞춰야 할 답, 난수로 초기화  static int top; //맞춰야 할 답의 범위에서 최고값  static int bottom; //맞춰야 할 답의 범위에서 최저값  static void init(){  srand((unsigned)time(0));  answer=rand()%100+1;  } // 난수를 생성하여 answer 변수로 초기화  static void msg(){  cout<<"답은 "<<bottom<<"과 "<<top<<"사이에 있습니다."<<endl;  } // top과 bottom 출력 public:  static void run(); // 게임 진행  static bool check(int answer){  if(answer<bottom || answer>top){  return false;  }  else if(answer==UpAndDownGame::answer){  return true;  }  else if(answer>UpAndDownGame::answer){  top=answer;  }  else if(bottom<answer) {  bottom = answer;  }   } //top과 bottom을 조절하고, 정답을 맞추었으면 true 리턴 };  bool Person::go() {  int num;  cout<<name<<">>";  cin>>num;  return UpAndDownGame::check(num);  } int UpAndDownGame::answer=0; int UpAndDownGame::top=100; int UpAndDownGame::bottom=1;  void UpAndDownGame::run() {  Person \*player[2];  string name;   cout << "Up & Down 게임을 시작합니다." << endl;  for(int i=0 ; i<2 ; i++){  cout<<"이름을 입력하세요 >> ";  cin>>name;  player[i]=new Person(name);  }   init();   int i=0;  while(true){  msg();  if(player[i]->go()){  cout<<player[i]->getName()<<"이 이겼습니다!!"<<endl;  break;  }  i++;  i%=2;  }  }   int main() {  UpAndDownGame::run(); } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 다음과 같은 static 멤버를 갖는 Random 클래스를 완성하고 제시된 결과처럼 출력될 수 있도록 main()함수도 작성하시오



class Random {

public:

static int nextInt(int min = 0, int max = 32767); //min과 max 사이의 랜덤 정수 리턴

static char nextAlphabet(); //랜덤하게 영문자 반환

static string nextString(int length); //매개변수로 전달된 길이를 갖는 임의의 문자열 반환

};

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <cstdlib> #include <ctime> #include <string> using namespace std;   class Random { public:  static void Srand() { srand( (unsigned)time(0) ); };  static int nextInt(int min = 0, int max = 32767){  int rand\_num;  rand\_num=rand()%max+min;  return rand\_num;  }; //min과 max 사이의 랜덤 정수 리턴  static char nextAlphabet(){  int rand\_num;  rand\_num=rand()%2;  if(rand\_num==0){  rand\_num=65+rand()%26;  return rand\_num;  }  else{  rand\_num=97+rand()%26;  return rand\_num;  }    }; //랜덤하게 영문자 반환  static string nextString(int length){  int num;  num=5+rand()%5;  cout<<num<<endl;  char a[num];  for(int i=0 ; i<num ;i++){  a[i]=Random::nextAlphabet();   }  string st=a;  if(st.size()>num){  st.erase(num); //길이를 넘어선 문자열이 생겨서 문자열 삭제코드 추가  }  return st;  } //매개변수로 전달된 길이를 갖는 임의의 문자열 반환 };  int main(){  Random::Srand();  cout<<"1에서 100까지 랜덤한 정수 10개를 출력합니다."<<endl;  for(int i=0 ; i<10 ; i++)  cout<<Random::nextInt(0,100)<<" ";  cout<<endl;  cout<<"알파벳을 랜덤하게 10개를 출력합니다."<<endl;  for(int i=0 ; i<10 ; i++)  cout<<Random::nextAlphabet()<<" ";  cout<<endl;  cout<<"길이가 5~10인 랜덤한 문자열 5개를 출력합니다."<<endl;  for(int i=0 ; i<5 ; i++) {  string st = Random::nextString(5);  cout << st << ": (" << st.size() << ")" << endl;  } } |
| [실행 결과] |

1. 다음과 같이 동일한 크기를 갖는 두 개의 배열을 매개변수로 전달하여 제시된 결과처럼 실행하는 프로그램을 완성 하시오.

class ArrayUtility2 {

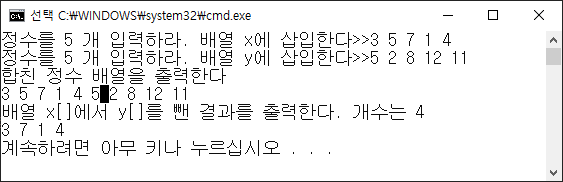
public:

static int\* concat(int s1[], int s2[], int size); // s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴

// s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴

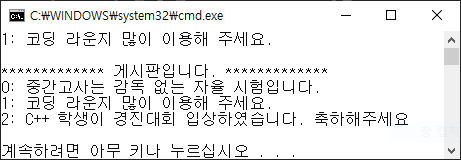
static int\* remove(int s1[], int s2[], int size, int& retSize);

};



|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class ArrayUtility2 { public:  static int\* concat(int s1[], int s2[], int size){  int\* p = new int[size];//동적 생성  int num = size/2;  for(int i=0; i<num; i++) {  p[i] = s1[i];  p[i+num] = s2[i];  }  return p;  } // s1과 s2를 연결한 새로운 배열을 동적 생성하고 포인터 리턴 // s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴  static int\* remove(int s1[], int s2[], int size, int& retSize){  int \*p;   for(int i=0 ; i<size/2 ;i++){  for(int j=0 ; j<size/2 ;j++){  if(s1[i]==s2[j]){  s1[i]=NULL;  }  }  }  p=new int[retSize];  int n=0;  for(int i=0 ; i<size/2 ; i++) {  if (s1[i] == NULL) {  retSize--;  continue;  }  p[n] = s1[i];  n++;   }  return p;  } };    int main(){  int x[5];  int y[5];  cout<<"정수를 5개 입력하라. 배열 x에 삽입한다.>>";  cin>>x[0]>>x[1]>>x[2]>>x[3]>>x[4];  cout<<"정수를 5개 입력하라. 배열 y에 삽입한다.>>";  cin>>y[0]>>y[1]>>y[2]>>y[3]>>y[4];  cout<<"합친 정수의 배열을 출력한다"<<endl;   int size=10;  int \*new\_arr1=ArrayUtility2::concat(x,y,size);  for(int i=0 ; i<size ; i++)  cout<<new\_arr1[i]<<" ";  cout<<endl;   int s1\_size=5;  int \*new\_arr2=ArrayUtility2::remove(x,y,size,s1\_size);  cout<<"배열 x[] 에서 y[]를 뺀 결과를 출력한다. 개수는"<<s1\_size<<endl;  for(int i=0 ; i<s1\_size ; i++)  cout<<new\_arr2[i]<<" "; } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 게시판 기능을 하는 Board 클래스를 활용하는 main()과 실행결과를 참고하여 Board 클래스를 만들고 전체 프로그램을 완성 하시오. 게시판 기능은 글을 올리는 기능과 게시글 모두를 출력하는 기능으로 제한한다.



int main() {

// Board myBoard; // 객체 생성은 컴파일 오류입니다.

Board::add("중간고사는 감독 없는 자율 시험입니다.");

Board::add("코딩 라운지 많이 이용해 주세요.");

Board::print();

Board::add("C++ 학생이 경진대회 입상하였습니다. 축하해주세요");

Board::print();

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Board{ public:  static string \*s;  static int max;  static int num;  static void add(string t){  num++;  s[num]=t;  }  static void print(){  if(num<2)  cout<<num<<": "<<s[num]<<endl;  else{  cout<<"\*\*\*\*\*\*\*\*\*\*\*게시판입니다. \*\*\*\*\*\*\*\*\*\*\*"<<endl;  for(int i=0 ; i<=num ; i++)  cout<<i<<": "<<s[i]<<endl;  }  } };  int Board::num=-1; int Board::max=10; string\* Board::s=new string[max];    int main() {  // Board myBoard; // 객체 생성은 컴파일 오류입니다.  Board::add("중간고사는 감독 없는 자율 시험입니다.");  Board::add("코딩 라운지 많이 이용해 주세요.");  Board::print();  Board::add("C++ 학생이 경진대회 입상하였습니다. 축하해주세요");  Board::print(); } |
| [실행 결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |