|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_13 | 학번 : | 20135151 | 이름 : | 이갑성 |

* 내용 점검 – 교재10장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 3번 |
| 2 | 3번 |
| 3 | 2번 |
| 4 | 2번 |
| 5 | template <class T>  bool equal(T a, T b)  {  if (a == b) {  return true;  }  else  return false;  } |
| 6 | template <class T>  void insert(T a, T b[], int index)  {  b[index] = a;  } |
| 7 | template <typename T>  T max(T x, T y) {  if (x > y)  return x;  else  return y;  } |
| 8 | template <class TYPE>  TYPE equals(TYPE x, TYPE y)  {  if (x == y)  return true;  else  return false;  } |
| 9 | (1)  int avg(int \*p, int n)  {  int k;  int sum = 0;  for (k = 0; k < n; k++)  sum += p[k];  return sum / n;  }  (2)  double avg(double \*p, int n)  {  int k;  double sum = 0;  for (k = 0; k < n; k++)  sum += p[k];  return sum / n;  } |
| 10 | (1) 예  (2) 3.14  (3) special 100 |
| 11 | 3번 |
| 12 | (1)  (2)  (3)  (4)  (5)  (6)  => 한번에 답변 작성  template <class T>  class Container {  T \*p;  int size;  public:  Container(int n) //생성자  {  p = new T[n];  }  ~Container() //소멸자  {  delete[] p;  }  void set(int index, T value) {  p[index] = value;  }  T get(int index)  {  return p[index];  }  };  int main()  {  Container<char> c(26);  for (int i = 0; i < 26; i++)  {  c.set(i, (char)('a' + i));  }  for (int i = 25; i >= 0; i--)  {  cout << c.get(i) << " ";  } |
| 13 | 1번 |
| 14 | (1) #include <vector> (2) #include <list>  (3) #include <merge> (4) #include <search> |
| 15 | vector<double> v;  v.push\_back(3.1);  v.push\_back(4.1);    for (int i = 0; i < v.size(); i++)  {  cout << v[i] << " ";  } |
| 16 | vector<char>::iterator it; |
| 17 | #include <map> |
| 18 | (1) vector  (2) map |

* 프로그램 과제

|  |
| --- |
| 1. 교재 527p : 문제9번 |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  using namespace std;  int main()  {  vector<int> v;  int num;  double sum = 0.0, ave;  bool flag = true;  do {  cout << "정수를 입력하세요(0을 입력하면 종료) >> ";  cin >> num;  sum += num;  if (num == 0)  flag = false;  v.push\_back(num);  for (int i = 0; i < v.size(); i++)  {  cout << v[i] << " ";  }  cout << endl;  ave = sum / v.size();  cout << "평균 = " << ave << endl;  } while (flag);  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 528p : 문제10번   Nation n[] = { Nation("미국", "와싱턴"), Nation("영국", "런던"), Nation("프랑스", "파리"),  Nation("중국", "베이찡"), Nation("일본", "도쿄"), Nation("러시아", "모스크바"),  Nation("브라질", "브라질리아"), Nation("독일", "베를린"), Nation("멕시코", "멕시코시티")};  Nation(수도명, 국가명) 이라는 객체를 만들어서 vector에 저장하기  vector<Nation> v;  =>Nation객체를 vector에 push\_back(n[i]);해준다.  퀴즈할 때 인덱스를 난수로 뽑아서 그 인덱스의 국가명이 나오게 한다.(v[index].getNationName();) |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  #include <string>  #include <time.h>  using namespace std;  class Nation {  string nation; //나라 이름  string capital; //수도  public:  Nation(string nation, string capital)  {  this->nation = nation;  this->capital = capital;  }  string getNation() //국가명 반환 메소드  {  return nation;  }  string getCapital() //도시명 반환 메소드  {  return capital;  }  };  int main()  {  vector<Nation> v;  int menu;  bool flag = true;  bool input;  string capital, nation;  int index;    Nation n[] = { Nation("미국", "와싱턴"), Nation("영국", "런던"), Nation("프랑스", "파리"),  Nation("중국", "베이찡"), Nation("일본", "도쿄"), Nation("러시아", "모스크바"),  Nation("브라질", "브라질리아"), Nation("독일", "베를린"), Nation("멕시코", "멕시코시티") };    for (int i = 0; i < 9; i++)  {  v.push\_back(n[i]);  }  cout << "\*\*\*\*\* 나라의 수도 맞추기 게임을 시작합니다. \*\*\*\*\*" << endl;  while (flag) {  cout << "정보입력: 1, 퀴즈: 2, 종료: 3 >> ";  cin >> menu;  switch (menu)  {  case 1:  cout << "현재 " << v.size() << "개의 나라가 입력되어 있습니다." << endl;  cout << "나라와 수도를 입력하세요(no no 이면 입력끝)" << endl;  for (;;) {  input = true;  cout << v.size() + 1 << ">>";  cin >> nation >> capital;  if (nation == "no" && capital == "no")  break;  for (int i = 0; i < v.size(); i++)  {  if (v[i].getNation() == nation)  {  cout << "already exist" << endl;  input = false;  }  }  if(input)  v.push\_back(Nation(nation, capital));  }  break;  case 2:  for (;;)  {  srand((unsigned)time(0));  index = rand() % v.size();  cout << v[index].getNation() << "의 수도는? ";  cin >> capital;  if (capital == "exit")  break;  if (v[index].getCapital() == capital)  {  cout << "Correct!!" << endl;  }  else  cout << "NO!!" << endl;  }  break;  case 3:  flag = false;  break;  }  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 529p : 문제12번   Word wordlist [] = { Word("love", "사랑"), Word("animal", "동물"), Word("human", "인간"),  Word("emotion", "감정"), Word("stock", "주식"), Word("trade", "거래"),  Word("society", "사회"), Word("baby", "자기"), Word("honey", "애인"),  Word("dall", "인형"), Word("bear", "곰"), Word("picture", "사진"), Word("painting", "그림"),  Word("fault", "오류"), Word("example", "보기"), Word("eye", "눈"), Word("statue", "조각상")}; |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  #include <string>  #include <time.h>  using namespace std;  class Word {  string eng;  string kor;  public:  Word(string eng, string kor)  {  this->eng = eng;  this->kor = kor;  }  string getEng()  {  return eng;  }    string getKor()  {  return kor;  }    };  int main()  {  int menu;  bool flag1 = true;  string eng, kor;  int index;  int munjeIndex = 0;  int collectIndex;  int select;  Word wordlist[] = { Word("love", "사랑"), Word("animal", "동물"), Word("human", "인간"),  Word("emotion", "감정"), Word("stock", "주식"), Word("trade", "거래"),  Word("society", "사회"), Word("baby", "자기"), Word("honey", "애인"),  Word("dall", "인형"), Word("bear", "곰"), Word("picture", "사진"), Word("painting", "그림"),  Word("fault", "오류"), Word("example", "보기"), Word("eye", "눈"), Word("statue", "조각상") };  vector<Word> v;  for (int i = 0; i < 17; i++)  {  v.push\_back(wordlist[i]);  }  cout << "\*\*\*\*\*\* 영어 어휘 테스트를 시작합니다. \*\*\*\*\*\*" << endl;    while (flag1)  {  cout << "영어 삽입: 1, 어휘 테스트: 2, 프로그램 종료: 그외키 >> ";  cin >> menu;    switch (menu)  {  case 1:  cout << "영어 단어에 exit을 입력하면 입력 끝" << endl;  while (true)  {  cout << "영어 >> ";  cin >> eng;  if (eng == "exit")  {  break;  }  cout << "한글 >> ";  cin >> kor;  v.push\_back(Word(eng, kor));  }  break;  case 2:  cout << "영어 어휘 테스트를 시작합니다. 1~4 외 다른 숫자 입력시 종료." << endl;  while (true) {  srand((unsigned)time(0));  index = rand() % v.size();  cout << v[index].getEng() << "?" << endl;  collectIndex = rand() % 4 + 1;  for (int i = 1; i < 5; i++)  {  munjeIndex = rand() % v.size();  if (collectIndex == i)  {  cout << "(" << i << ") " << v[index].getKor() << " ";  }  else  cout << "(" << i << ")" << v[munjeIndex].getKor() << " ";  }  cout << ": >> ";  cin >> select;  if (!(select >= 1 && select <= 4))  {  break;  }  if (select == collectIndex)  {  cout << "Excellent!!" << endl;  }  else  cout << "No!!" << endl;    }  break;  default:  flag1 = false;  break;  }  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 530p : 문제14번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <map>  using namespace std;  int main()  {  string id, pass;  bool flag = true;  map<string, string> info;  int menu;  cout << "\*\*\*\*\* 암호 관리 프로그램 WHO를 시작합니다. \*\*\*\*\*" << endl;  while (flag)  {  cout << "삽입: 1, 검사: 2, 종료: 3 >> ";  cin >> menu;  switch (menu)  {  case 1:  cout << "이름 암호 >> ";  cin >> id >> pass;  info.insert(make\_pair(id, pass));  break;  case 2:  cout << "이름? ";  cin >> id;  while (true) {  cout << "암호? ";  cin >> pass;  if (info[id] == pass)  {  cout << "통과!!" << endl;  break;  }  else  {  cout << "실패~~" << endl;  }  }  break;  case 3:  cout << "프로그램을 종료합니다...." << endl;  flag = false;  break;  }  }    } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 531p : 문제15번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <iterator>  #include <vector>  using namespace std;  class Circle {  string name; //이름  int radius; //반지름  public:  Circle(int radius, string name) {  this->radius = radius;  this->name = name;  }  double getArea() {  return 3.14 \* radius \* radius;  }  string getName() {  return name;  }  };  int main()  {  vector<Circle> v;  vector<Circle>::iterator it;  int menu, r;  bool flag = true;  string name;  cout << "원을 삽입하고 삭제하는 프로그램입니다." << endl;  while (flag)  {  cout << "삽입: 1, 삭제: 2, 모두보기: 3, 종료: 4 >> ";  cin >> menu;  switch (menu)  {  case 1:  cout << "생성하고자 하는 원의 반지름과 이름은 >> ";  cin >> r >> name;  v.push\_back(Circle(r, name));  break;  case 2:  cout << "삭제하고자 하는 원의 이름은 >> ";  cin >> name;  for (it = v.begin(); it != v.end(); it++)  {  if (it->getName() == name)  {  it = v.erase(it);  if (it == v.end())  {  cout << "더이상 가르킬게 없습니다." << endl;  break;  }  }  }    break;  case 3:  for (int i = 0; i < v.size(); i++)  {  cout << v[i].getName() << endl;  }  break;  case 4:  flag = false;  break;  }  }  } |
| **[실행결과]** |