|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_13 | 학번 : | 20145128 | 이름 : | 박인근 |

* 내용 점검 – 교재10장 연습문제

|  |  |
| --- | --- |
| 번호 | 문제풀이 |
| 1 | 3 |
| 2 | 3 |
| 3 | 2 |
| 4 | 2 |
| 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;  }  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>  bool 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;  }   1. 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) 공존할 수 있다.  (2) 3.14 이유는 템플릿 함수가 호출 되었기 때문  (3) special 100 템플릿 함수가 아닌 일반 함수가 호출 되었기 때문 |
| 11 | 3 |
| 12 | 1. T \*p   int size;   1. template <class T> Container<T>::Container(int n){   p = new T[n];  size = n;  }   1. template <class T> Container<T>::~Container(){   delete[] p;  }   1. template <class T> T Container<T>::get(int index){   return p[index];  }   1. Container<char> c(26); 2. int main() {   Container <char> c(26);  for(int i = 0 ; i < 26 ; i ++)  c.set(i, ‘a’+i);  for(int i =25; i >= 0; i--)  cout << c.get(i);  } |
| 13 | 1 |
| 14 | 1. <vector> 2. <list> 3. <algorithm> 4. <algorithm> |
| 15 | vector<double> v;  for(int i = 0; i < v.size(); i++){  cout << v[i] << endl;  } |
| 16 | vector<char>::iterator it; |
| 17 | #include <map> |
| 18 | 1. verctor 시간별로 계속 읽어 저장해야 하기 때문에 가변배열인 vector 가 더 유리 2. map 지역별로 해가 뜨는 시간을 묶어서 저장하면 편하므로 map 이 유리 |

* 프로그램 과제

|  |
| --- |
| 1. 교재 527p : 문제9번 |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  #include <string>  using namespace std;  int main() {  vector<int> sv; // 정수형 벡터 생성  int p, sum = 0, count = 0;  while (1) {  cout << "정수를 입력하세요(0을 입력하면 종료)>>";  cin >> p;  if (p == 0) break;  sv.push\_back(p);  for (int i = 0; i < sv.size() ; i++) {  cout << sv[i] << ' ';  sum += sv[i];  }  cout << "\n평균 = " << double(sum)/sv.size() << endl;  sum = 0;  }  cout << endl;  system("pause");  } |
| **[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")}; |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  #include <string>  #include <ctime>  using namespace std;  class Nation {  string ctry;  string cname;  public:  Nation(string ctry, string cname) {  this->ctry = ctry;  this->cname = cname;  }  string getCtry() { return ctry; }  string getCName() { return cname; }  };  int main() {  int random;  vector<Nation> v;  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") };  v.assign(n, n + 9);  int select,flag;  string ctry, cname;    cout << "\*\*\*\*\* 나라의 수도 맞추기 게임을 시작합니다. \*\*\*\*\*" << endl;  while (1) {  srand((unsigned int)time(NULL));  cout << "정보 입력: 1, 퀴즈: 2, 종료: 3 >> ";  cin >> select;  if (select == 1) {  cout << "현재 " << v.size() << "개의 나라가 입력되어 있습니다." << endl;  cout << "나라와 수도를 입력하세요(no no 이면 입력 끝)" << endl;  while (1) {  flag = 0;  cout << v.size() + 1 << ">>";  cin >> ctry >> cname;    if (ctry == "no" && cname == "no") break;    for (int i = 0; i < v.size(); i++) {  if (ctry == v[i].getCtry()) {  cout << "already exists !!" << endl;  flag = 1;  break;  }  }  if (flag == 1) continue;  v.push\_back(Nation(ctry, cname));  }  }  else if (select == 2) {  for (int i = 0; ; i++) {  random = rand() % v.size();  cout << v[random].getCtry() << "의 수도는?";  cin >> cname;  if (cname == "exit") break;  if (cname == v[random].getCName()) cout << "Correct !!" << endl;  else cout << "NO !!" << endl;  }  }  else if (select == 3) break;  }    getchar();  } |
| **[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 <ctime>  using namespace std;  class Word {  string korean, english;  public:  Word(string english, string korean) {  this->english = english;  this->korean = korean;  }  string getKorean() { return korean; }  string getEnglish() { return english; }  };  int main()  {  int r, select, n[4],key;  string ko[4], collect , eg, k;  for (int i = 0; i < 4; i++) ko[i] = "";  for (int i = 0; i < 4; i++) n[i] = 0;  vector<Word> v;  srand((unsigned int)time(NULL));  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") };  v.assign(wordlist, wordlist + 17);  cout << "\*\*\*\*\* 영어 어휘 테스트를 시작합니다. \*\*\*\*\*" << endl;    while (1) {  cout << "어휘 삽입: 1, 어휘 테스트 : 2, 프로그램 종료:그외키 >> ";  cin >> key;  if (key == 1) {  cout << "영어 단어에 exit을 입력하면 끝" << endl;  while (getchar() != '\n');  while (1) {  cout << "영어 >>";  cin >> eg;  if (eg == "exit") break;  cout << "한글 >>";  cin >> k;  v.push\_back(Word(eg, k));  }  }  else if (key == 2) {  cout << "영어 어휘 테스트를 시작합니다. 1~4 외 다른 입력시 종료합니다." << endl;  while (1) {  r = rand() % v.size();  cout << v[r].getEnglish() << "?" << endl;  ko[0] = v[r].getKorean();  collect = ko[0];  for (int i = 1; i < 4; i++) {  r = rand() % v.size();  ko[i] = v[r].getKorean();  }  n[0] = rand() % 4;  while (1) {  for (int i = 1; i < 4; i++) {  n[i] = rand() % 4;  }  if (n[0] != n[1] && n[0] != n[2] && n[0] != n[3] && n[1] != n[2] && n[1] != n[3] && n[2] != n[3]) break;  }  while (1)  {  for (int i = 1; i < 4; i++) {  r = rand() % v.size();  ko[i] = v.at(r).getKorean();  }  if (ko[0] != ko[1] && ko[0] != ko[2] && ko[0] != ko[3] && ko[1] != ko[2] && ko[1] != ko[3] && ko[2] != ko[3]) break;  }  for (int i = 0; i < 4; i++) {  cout << "(" << i + 1 << ") " << ko[n[i]] << ' ';  }  cout << ":>";  cin >> select;  if (select < 1 || select > 4) {  break;  }  if (ko[n[select - 1]] == collect) {  cout << "Excellent !!" << endl;  continue;  }  else {  cout << "No. !!" << endl;  continue;  }  }  }  else {  cout << "프로그램 종료!!" << endl;  break;  }      }  system("pause");  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 530p : 문제14번 |
| **[프로그램 소스]**  #include <iostream>  #include <map>  #include <string>  using namespace std;  int main()  {  map<string, string> dic;  int key;  string name, pw;  cout << "\*\*\*\*\* 암호 관리 프로그램 WHO를 시작합니다. \*\*\*\*\*" << endl;  while (1) {  cout << "삽입:1, 검사:2, 종료:3>> ";  cin >> key;  if (key == 1) {  cout << "이름 암호>> ";  cin >> name >> pw;  dic.insert(make\_pair(name, pw));  }  else if (key == 2) {    cout << "이름? ";  cin >> name;  while (1) {  cout << "암호? ";  cin >> pw;  if (pw == dic[name]) {  cout << "통과!!" << endl;  break;  }  else {  cout << "실패~~" << endl;  continue;  }  }  }  else if (key == 3) { cout << "프로그램을 종료합니다..." << endl; break; }  }  getchar();  getchar();  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 531p : 문제15번 |
| **[프로그램 소스]**  #include <iostream>  #include <vector>  #include <string>  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()  {  int key, radius;  string name;  vector<Circle \*> v;  vector<Circle \*>::iterator it;  cout << "원을 삽입하고 삭제하는 프로그램입니다." << endl;  while (1) {  cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ";  cin >> key;  if (key == 1) {  cout << "생성하고자 하는 원의 반지름과 이름은 >> ";  cin >> radius >> name;  v.push\_back(new Circle(radius, name));  continue;  }  else if (key == 2) {  cout << "삭제하고자 하는 원의 이름은 >> ";  cin >> name;  it = v.begin();  while (it != v.end()) {  Circle \*p = \*it;  if (p->getName() == name) {  it = v.erase(it);  delete p;  }  else it++;  }  continue;  }  else if (key == 3) {  for (it = v.begin() ; it != v.end() ; it++) {  cout << (\*it)->getName() << endl;  }  cout << endl;  continue;  }  else if (key == 4) {  cout << "프로그램을 종료합니다..." << endl;  break;  }  }    getchar();  getchar();  } |
| **[실행결과]** |