|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_15 | 학번 : | 20157135 | 이름 : | 이상우 |

* getline() 으로 파일 읽고 단어 검색

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <string>  #include <vector>  using namespace std;  void fileRead(vector<string> &v, ifstream &fin) { // fin 스트림으로부터 벡터 v에 읽어들임  string line;  while(getline(fin, line)) { // fin 파일에서 한 라인 읽기  v.push\_back(line); // 읽은 라인을 벡터에 저장  }  }  void search(vector<string> &v, string word) { // 벡터 v에서 word를 찾아 출력  for(int i=0; i<v.size(); i++) {  int index = v[i].find(word);  if(index != -1) // found  cout << v[i] << endl;  }  }  int main() {  vector<string> wordVector;  ifstream fin("words.txt");  if(!fin) {  cout << "words.txt 파일을 열 수 없습니다" << endl;  return 0; // 열기 오류  }  fileRead(wordVector, fin); // fin 스트림으로부터 wordVector에 라인 별로 읽기  fin.close();  cout << "words.txt 파일을 읽었습니다." << endl;  while(true) {  cout << "검색할 단어를 입력하세요 >>";  string word;  getline(cin, word); // 키보드로부터 문자열 읽기  if(word == "exit")  break; // 프로그램 종료  search(wordVector, word); // 문자열을 words.txt에서 검색하여 출력  }  cout << "프로그램을 종료합니다." << endl;  } |
| **[실행 결과]** |

* 바이너리 I/O - 블록단위 입출력

|  |
| --- |
| #include <iostream>  #include <fstream>  using namespace std;  int main() {  const char\* file = "c:\\temp\\data.dat";  ofstream fout;  fout.open(file, ios::out | ios::binary); // 읽기 모드로 파일 열기  if(!fout) { // 열기 실패 검사  cout << "파일 열기 오류";  return 0;  }  int n[] = {0,1,2,3,4,5,6,7,8,9};  double d = 3.15;  fout.write((char\*)n, sizeof(n)); // int 배열 n을 한 번에 파일에 쓴다.  fout.write((char\*)(&d), sizeof(d)); // double 값 하나를 파일에 쓴다.  fout.close();  // 배열 n과 d 값을 임의의 값으로 변경시킨다.  for(int i=0; i<10; i++) n[i]=99;  d = 8.15;  // 배열 n과 d 값을 파일에서 읽어 온다.  ifstream fin(file, ios::in);  if(!fin) { // 열기 실패 검사  cout << "파일 열기 오류";  return 0;  }  fin.read((char\*)n, sizeof(n)); // int 배열을 한 번에 읽오 온다.  fin.read((char\*)(&d), sizeof(d)); // double 값을 읽어 온다.  for(int i=0; i<10; i++) // 읽은 배열 n을 확인한다.  cout << n[i] << ' ';  cout << endl << d << endl; // 읽은 double 값을 확인한다.  fin.close();  } |
| **[실행 결과]** |

* 프로그램 과제

1. 교재 639p : 문제 7번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <fstream>  #include <string>  #include <vector>  using namespace std;  int main() {  char c; long size = 0;  int index = 0;  char \*ch;  ifstream fin("system.ini", ios::in | ios::binary); // 읽기 모드로 파일 열기  ofstream fout("system.txt", ios::out | ios::binary); // 읽기 모드로 파일 열기  ifstream fin2("system.txt", ios::in | ios::binary); // 읽기 모드로 파일 열기  ofstream fout2("copy\_system.txt", ios::out | ios::binary); // 읽기 모드로 파일 열기  if (!fin) {  cout << "system.ini 파일을 열 수 없습니다" << endl;  c = getchar();  return 0; // 열기 오류  }else if(!fout) {  cout << "system.txt 파일을 열 수 없습니다" << endl;  c = getchar();  return 0; // 열기 오류  }else {  fin.seekg(0, std::ios::end);  size = fin.tellg();  fin.seekg(0, ios::beg);  ch = new char[size];  cout << "===원본 파일을 읽습니다.==="<<endl;  while ((c=fin.get())!=EOF) {  ch[index] = c;  cout <<ch[index];  index++;  }fin.close();  cout << "\n===파일을 거꾸로 출력합니다.===\n";  for (int i = size-1; i >= 0; i--) {  fout << ch[i];  cout << ch[i];  }  }  fin.close();  fout.close();  cout << "\n\n ===test 를 진행합니다.===\n";  index = 0;  if (!fin2) {  cout << "system.txt 파일을 열 수 없습니다" << endl;  c = getchar();  return 0; // 열기 오류  }  else if (!fout2) {  cout << "copy\_system.txt 파일을 열 수 없습니다" << endl;  c = getchar();  return 0; // 열기 오류  }  else {  cout << "\n===꺼꾸로 저장한 파일을 읽습니다..===\n";  while ((c = fin2.get()) != EOF) {  ch[index] = c;  cout << ch[index];  index++;  }  cout << "\n\n===파일을 거꾸로 출력 합니다.===\n\n";  for (int i = size - 1; i >= 0; i--) {  fout2 << ch[i];  cout << ch[i];  }index = 0;  }  fin2.close();  fout2.close();  c = getchar();  } |
| **[실행 결과]** |

1. 교재 641p : 문제 13번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <fstream>  #include <string>  #include <vector>  using namespace std;  void find(string word, vector<string> &v)  {  int findCount = 0; // 단어 존재 여부를 체크하는 카운트 변수  for (int i = 0; i < v.size(); i++) {  int index = v[i].find(word);  // 벡터의 요소에서 해당 단어를 포함하는 지 검사한다.  // 없는 경우에는 -1이 반환 되며, 포함된 경우 해당 위치(인덱스)를 반환한다.  if (index == 0) { // 첫번 째에 위치한 경우  findCount++;  cout << v[i] << endl;  }  }  if (!findCount) cout << "발견할 수 없음!" << endl;  }  int main() {  vector<string> wordVector; // vector<string> 선언  ifstream fin;  fin.open("words.txt"); // words.txt 파일 오픈  if (!fin) {  cout << "words.txt 파일을 열 수 없습니다." << endl;  return 0;  }  string line;  while (true) {  getline(fin, line);  if (fin.eof()) break; // 끝까지 읽기  wordVector.push\_back(line); // 벡터에 읽은 단어 삽입  }  cout << " ...words.txt 파일 로딩 완료" << endl;  fin.close();  cout << "검색을 시작합니다. 단어를 입력해 주세요." << endl;  while (true) {  cout << "단어>>";  string word;  getline(cin, word); // getline을 이용한 문자열 입력  if (word == "exit") {  break;  }  else {  find(word, wordVector);  }  }  } |
| **[실행 결과]** |