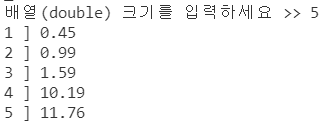
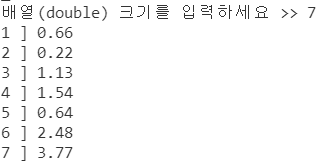
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_07\_스마트 포인터와 문자열 | 학번 : | 20175260 | 이름 : | 최영빈 |

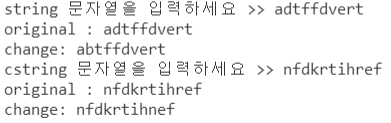
* **개념 적용 응용 프로그래밍 – 스마트 포인터와 문자열**

1. 사용자로부터 배열 크기를 입력 받아 실수 배열을 동적 할당하여 난수로 초기화 한 후 소수 이하 둘째자리까지 출력하는 프로그램을 작성하시오. 단, 스마트 포인터를 사용하시오.

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <memory>  #include <ctime>  #include <cstdlib>  #include <iomanip>  using namespace std;  int main()  {  srand(time(nullptr));  int size;  cout << "배열 (double) 크기를 입력하세요 >>";  cin >> size;  auto arr = make\_unique<double[]>(size);  for(int i =0; i <size; i++){  arr[i] = rand()%10+(double)rand()/(RAND\_MAX);  }  for(int i = 0; i< size; i++){  cout << i+1 << " ] " <<fixed<<setprecision(2)<<arr[i]<<endl;  }  return 0;  } |
| [실행 결과]  이미지  이미지 |

1. string 클래스를 이용하여 사용자가 입력한 영문 한 줄을 입력 받고 글자 하나만 랜덤하게 수정하여 출력하는 프로그램을 완성하시오.



//매개변수로 받은 문자열에서 글자 하나만 랜덤하게 수정하여 수정된 문자열 반환

string changeExt(string\_view text);

int main()

{

  string text;

  char str[100];

  cout<<"string 문자열을 입력하세요 >> ";

  cout<<"change: " << changeExt(text)<<endl;

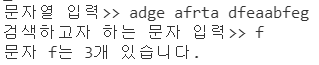
  cout<<"cstring 문자열을 입력하세요 >> ";

    cout<<"change: " << changeExt(str)<<endl;

}

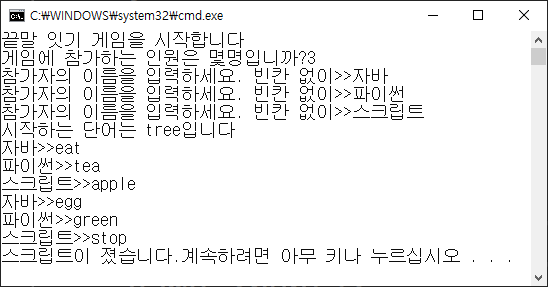
|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string\_view>  #include <string>  #include <ctime>  using namespace std;  int value;  int ch;  string changeExt(string\_view text)  {  srand(time(nullptr));  value = rand()%122+97;  ch = rand()%text.size()-1;  string str = text.data();  char c1 = value;  string c2;  c2.push\_back(c1);  str = str.replace(ch, 1, c2);  return str;  }  int main()  {  string text;  char str[100] = "oohello";  cout << "string 문자열을 입력하세요 >> ";  cin >> text;  cout << "Original: " << text << endl;  cout << "change: " << changeExt(text) << endl;  cout << "cstring 문자열을 입력하세요 >> ";  cin >> str;  cout << "Original: " << str << endl;  cout << "change: " << changeExt(str) << endl;  } |
| [실행 결과]  이미지 |

1. 빈칸을 포함하는 문자열을 입력 받고 문자열에서 입력 받은 문자가 몇 개 있는지 출력하는 프로그램을 작성하시오. 단, 문자열에서 특정 문자 검색은 string 클래스의 find() 멤버 함수를 이용하시오.



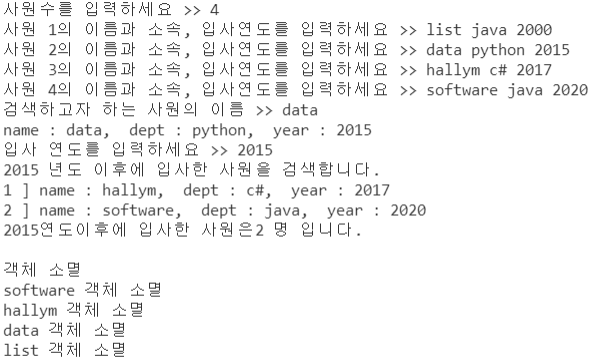
|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <memory>  using namespace std;  int main()  {  string name;  char search;  int count;  cout<< "문자열 입력>>";  getline(cin,name);  cout<<"검색하고자 하는 문자 입력 >>";  cin >> search;  for(int i = 0; i < name.size();i++){  if([name.at](http://name.at)(i) == search)  count++;  }  cout <<"문자 " << search<< "는 "<< count<<"개 있습니다." <<endl;  } |
| [실행 결과]  이미지 |

1. 제시된 결과처럼 끝말잇기 게임을 실행하는 프로그램을 작성하시오. 단, 끝말 잇기 게임에 참여하는 선수의 수를 입력 받고 선수 각 사람의 이름을 입력 받은 후 게임을 시작한다. 단, 메모리할당은 스마트 포인터를 사용하시오



|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <memory>  #include <vector>  using namespace std;  int main(){  int size;  string name;  string v1 = "tree";  string v2;  bool flag = true;  cout << "끝말 잇기 게임을 시작합니다 "<<endl;  cout <<" 게임에 참가하는 인원은 몇명입니까? "<<endl;  cin >> size;  auto arr = make\_unique<string[]>(size);  for(int i =0; i < size; i++){  cout <<"참가자의 이름을 입력하세요. 빈칸 없이 >>";  cin >>name;  arr[i] = name;  }  cout << "시작하는 단어는 "<<v1<<"입니다."<<endl;    while(flag){  for(int i =0 ; i < size; i++){  cout << arr[i] <<" >> ";  cin >>v2;  if(v1.substr(v1.size()-1) == v2.substr(0,1)){  v1 = v2;  continue;  }  else{  cout << arr[i] <<"이 졌습니다. 계속하려면 아무 키나 누르십시오 ..."<<endl;  flag = false;  break;  }  }  }  return 0;  } |
| [실행 결과]  이미지 |

1. 다음은 이름과 소속, 입사 연도를 속성으로 갖는 Employee 클래스와 이들을 배열로 관리하는 Manager클래스이다. 키보드에서 직원 수를 입력 받고, 그 개수만큼 이름과 소속, 입사 연도를 입력 받고 다음과 같이 실행되도록 main()함수를 작성하시오. Employee 클래스와 Manager 클래스도 완성 하시오



class Employee

{

  int year;    // 입사연도

  string name; //이름

  string dept; //소속

public:

  void setEmployee(string name, string dept, int year); //입사연도, 이름, 소속을 설정

  int getYear();                                        //입사연도 반환

  string getName();                                     //이름 반환

  string getDept();                                     //소속 반환

  void disPlay();                                       //개인정보 출력

  ~Employee();                                         //소멸자

};

class Manager

{

  unique\_ptr<Employee[]> p; // Employee 배열에 대한 포인터

  int size;                 // 배열의 크기

public:

  Manager(int size); // size 크기의 배열을 동적 생성. 사용자로부터 입력 완료

  ~Manager();

  void searchByName(); // 사용자로부터 원의 이름을 입력받아 소속과 입사연도 출력

  void searchByYear(); // 사용자로부터 입력받은 입사연도와 일치하는 값을 갖는 직원정보 출력

};

int main()

{

  Manager \*pManager;

  cout << "사원수를 입력하세요 >> ";

  int size;

  cin >> size;

  if (size <= 0)

  {

    cout << "양수를 입력하세요." << endl;

    return 0;

  }

//나머지 코드는 본인 작성

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <memory>  using namespace std;  class Employee{  int year;  string name;  string dept;  public:  Employee(){;}  void setEmployee(string name,string dept, int year);  int getYear(){return year;}  string getName(){return name;}  string getDept(){return dept;}  void disPlay();  ~Employee();  };  void Employee::setEmployee(string name, string dept, int year){  this->name = name;  this->dept = dept;  this->year = year;  }  void Employee::disPlay(){  cout << "name :"<<name<<" dept : "<<dept<<" year : "<<year<<endl;  }  Employee::~Employee(){  cout <<name <<" 객체 소멸"<<endl;  }  class Manager{  unique\_ptr<Employee[]> p;  int size;  string name;  string dept;  int year;  public:  Manager(int size);  ~Manager();  void searchByName();  void searchByYear();  };  Manager::Manager(int size){  this->size = size;  p = make\_unique<Employee[]>(size);  for (int i = 0; i < size; i++)  {  cout << "사원 " << i + 1 << "의 이름과 소속, 입사연도를 입력하세요 >>";  cin >> name;  cin >> dept;  cin >> year;  p[i].setEmployee(name,dept,year);  }  }  void Manager::searchByName(){  cout << "검색하고자 하는 사원의 이름 >>";  cin >> name;  for(int i = 0; i < size; i ++){  if(p[i].getName() == name){  p[i].disPlay();  }  }  }  void Manager::searchByYear(){  int count = 0;  cout << "입사 연도를 입력하세요 >>";  cin >> year;  cout <<year <<"이후에 입사한 사원을 검색합니다. "<<endl;  for(int i = 0; i < size; i ++){  if(p[i].getYear() > year){  cout << i+1 <<" ] ";  p[i].disPlay();  count++;  }  }  cout << year<<"이후에 입사한 사원은 "<<count<<" 명 입니다."<<endl;  }  Manager::~Manager(){  cout << "객체 소멸"<<endl;  }  int main()  {  int size;  Manager \*pManager;  cout << "사원수를 입력하세요 >>";  cin >> size;  if(size <= 0){  cout << "양수를 입략하세여."<<endl;  return 0;  }  pManager = new Manager(size);  pManager->searchByName();  pManager->searchByYear();  delete pManager;  return 0;  } |
| [실행 결과]  이미지 |