**2017, 봄 학기, 고급 객체지향 프로그래밍 시험 1**

1. **다음 검색 프로그램을 구현하라. (25점)**
   1. 다음 조건에 맞게 길이가 가변적인 2차원 배열을 동적 할당하라.
      1. 사용자로부터 4개의 정수형 변수를 받아라. 4개의 수는 각각 행의 최소, 최대, 열의 최소, 최대 개수를 나타낸다. (단, 행과 열의 최소 개수는 2이상의 값이 입력되며 최대값은 항상 최소값보다 크다고 가정)
      2. 먼저, 행의 최대 최소 사이의 랜덤 수를 생성하여 행의 개수를 결정하고, 각 행마다 **개별적으로** 열의 개수 역시 최소 최대를 감안하여 랜덤하게 결정한 후 이 길이가 가변적인 2차원 배열의 동적 할당 수행
      3. 각 생성된 2차원 배열의 내용을 1~1000 사이의 랜덤값으로 초기화.
      4. **배열 안의 값을 화면에 출력**
   2. 키 값을 하나 입력 받고, 위에서 생성한 2차원 배열의 원소 중, 해당 키값이 가장 먼저 일치하는 원소를 검색하여 그 원소의 **행과 열을 참조 변수(pass by reference)**로 넘겨주는 함수를 구현하라. **main함수에서 해당 행과 열의 값을 출력**하라.
   3. 그 후 **동적할당한 배열을 모두 해제**하라. (**해제 시에 비정상 종료되면 실행 오류로 간주하여 0점 처리함.**)

구현에 필요한 변수, 함수 및 자료구조 등은 자유롭게 추가하거나 수정하여 사용하라.

예) 행, 열의 최대최소가 4, 6 3, 7이고, 사용자로부터 입력 받은 검색키 값이 48일 경우.

행의 최소최대개수, 열의 최소최대개수를 각각 입력하시오. 4 6 3 7

배열의 생성 예

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 53 | 48 | 781 | 159 | 268 | 348 |  |
| 561 | 224 | 108 |  |  |  |  |
| 339 | 857 | 907 | 356 |  |  |  |
| 627 | 644 | 549 | 915 | 341 | 351 | 366 |

찾을 키값을 입력하시오. 48

해당 자료는 1행 2열에 있습니다..

만약 검색한 키값이 존재하지 않는다면 다음과 같이 출력.

해당 자료는 존재하지 않습니다.

1. **다음 토너먼트 승부 프로그램을 구현하라. (25점)**
   1. 사용자로부터 총 참여자수를 입력 받아 진행될 회전수 계산. ex) 참여자가 8명이면 3회전 진행, 16명이면 4회전 진행 등등.

(단, 참여자수는 2의 거듭제곱수로 입력됨을 가정)

* + 1. int \*\* type 변수에 int\* type으로 회전수+1개만큼 동적 할당.
    2. 위에서 동적할당한 int \* 배열의 첫 번째 원소에 참여자수만큼 int 배열을 동적 할당. (첫번째 행)
    3. 첫번째 행의 각 원소 값을 1~1000 사이의 랜덤값으로 초기화.
    4. 첫번째 행의 원소들을 화면에 출력.
  1. 그 후 아래 과정을 최종 우승자가 정해질 때까지 반복. (다음과 같은 함수로 구현 추천)

bool tournament(int \*\*arr, int stageNum, int playerNum, int &currentStage);

* + 1. 다음 행에 현재 행의 절반 크기로 int 배열을 동적 할당.
    2. 좌측부터 각각 2개 원소를 비교하여 수가 큰 쪽이 승리하는 방식으로 토너먼트를 진행하여 승리자를 다음 행에 저장.
    3. 다음 행의 내용을 화면에 출력.
  1. 모든 수행을 마친 후 **동적할당한 배열을 모두 해제**.

. (**해제 시에 비정상 종료되면 실행 오류로 간주하여 0점 처리함.**)

구현에 필요한 변수, 함수 및 자료구조 등은 자유롭게 추가하거나 수정하여 사용하라.

예) 사용자 입력 8

배열 생성 예

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 초기단계 | 361 | 257 | 772 | 109 | 983 | 311 | 396 | 484 | 1회전\_진행후 | 361 | 257 | 772 | 109 | 983 | 311 | 396 | 484 |
|  |  |  |  |  |  |  |  |  |  | 361 | 772 | 983 | 484 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2회전\_진행후 | 361 | 257 | 772 | 109 | 983 | 311 | 396 | 484 | 3회전\_진행후 | 361 | 257 | 772 | 109 | 983 | 311 | 396 | 484 |
|  | 361 | 772 | 983 | 484 |  |  |  |  |  | 361 | 772 | 983 | 484 |  |  |  |  |
|  | 772 | 983 |  |  |  |  |  |  |  | 772 | 983 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | 983 |  |  |  |  |  |  |  |

1. **연락처 관리 프로그램 MemberContact를 만들려고 한다. MemberContact에서 쓰일 연락처 정보저장 클래스 Contact 클래스를 구현하고 테스트하라. (25점)**
   1. Contact 클래스의 멤버 변수 (세 변수 모두 문자형 배열로 선언)
      1. char name[20]: 친구 이름
      2. char hphone[20]: 핸드폰 번호
      3. char email[30]: 이메일 주소
   2. 멤버 함수
      1. 기본 생성자: 기본 매개변수 값(이름은 홍길동, 핸드폰 번호는 010-1234-1234으로, 이메일 주소는 [noname@khu.ac.kr](mailto:noname@khu.ac.kr))을 입력
      2. void setName(char\*): 파라미터로 받은 이름으로 이름 바꾸기
      3. void setPhone(char\*): 파라미터로 받은 번호로 번호 바꾸기
      4. void setEmail(char\*): 파라미터로 받은 이메일로 이메일 바꾸기
      5. void printContact(): 현재 연락처의 내용을 출력해 주는 함수
   3. 위 클래스를 구현한 다음 main 함수에서 다음과 같은 일을 수행하라. 예상 결과는 오른쪽과 같음  
       \* 클래스 구현 시 멤버 변수와 함수들을 private, public으로 적절히 구분해 구현해야 함.

**힌트: 문자열 복사를 위해 strcpy 혹은 strcpy\_s 사용. string.h 를 include 해야 함**

**사용법은**

**strpcy(복사받을 문자열 포인터, 원본 문자열 포인터)**

**strcpy\_s(복사받을 문자열 포인터, 문자열 길이, 원본 문자열 포인터)**

1. **복소수(실수부, 허수부) 클래스 Complex를 구현하고 테스트 하라. (25점)**
   1. Complex 클래스의 멤버 변수
      1. double real: 실수부
      2. double image: 허수부
   2. 멤버 함수
      1. 기본 생성자: 기본 매개변수 값(실수부,허수부 각각 0.0) 을 입력
      2. double getReal(): 실수부 데이터 가져오기
      3. double getimage(): 허수부 데이터 가져오기
      4. void setReal(double): 파라미터로 받은 실수부로 실수부 데이터 바꾸기
      5. void setImage(double): 파라미터로 받은 허수부로 허수부 데이터 바꾸기
      6. void print(): 실수부와 허수부를 지정된 형태(실수, 허수)로 출력
   3. 멤버 함수로 복소수의 덧셈 (add), 곱셈 (multi)을 구현하라.
      1. x1(x1,y1), x2(x2,y2)인 경우 복소수 add는 x3=x1+x2, y3 = y1+y2,
      2. x1(x1,y1), x2(x2,y2)인 경우 복소수 multi는 x3=x1\*x2 - y1\*y2, y3 = x1\*y2 + y1\*x2
      3. 두 함수는 Complex 타입 파라미터를 하나 받아 자신과 파라미터 사이의 해당 연산을 수행 후 결과를 자신에게 넣고 또한 이를 리턴 해 줌.
   4. 외부 함수로 위의 뎃셈, 뺄셈을 구현하라. 두 함수를 위 클래스의 프렌드 함수로 선언하여 구현하라. 두 함수는 두개의 복소수를 파라미터로 받아 이 둘의 연산 결과를 리턴해 주는 함수.
   5. 위 맴버 변수들과 함수들을 적절히 구현하여 다음과 같은 Main 함수의 코드가 동작하여 다음과 같은 결과를 화면에 출력하게 만들어라.

**int main() {  
 Complex c1(3,5),c2(7),c3;**

**c3=add(c1,c2);**

**cout<<"c1 : "; c1.print();**

**cout<<"c2 : "; c2.print();**

**cout<<"c3=c1+c2"<<endl;**

**cout<<"c3 : "; c3.print();**

**Complex x(1,2),y(2),z;**

**z=multi(x,y);**

**cout<<"x : "; x.print();**

**cout<<"y : "; y.print();**

**cout<<"z=x\*y"<<endl;**

**cout<<"z : "; z.print();**

**//추가로, 여기에 멤버함수로 만든 add와 multi 를 위와 같이 테스트할 수 있는 코드를 작성하라.**

**return 0;**

**}**