**자료구조와 알고리즘 보고서**

**1. 이진수를 십진수로 변환하는 함수**

**소스코드**

#include <iostream>

#include <string>

using namespace std;

bool validateBinary(int num);

int binaryToDecimal(int num);

int main() {

int num;

int decimal;

cout << "20215229 이채훈" << endl;

while (true) {//이진수 나올때까지 계속 반복

cout << endl;

cout << "이진수(1 또는 0): ";

cin >> num;

if (validateBinary(num)) {

decimal = binaryToDecimal(num);

cout << endl;

cout << "입력한 이진수: " << num << endl;

cout << "변환 된 십진수 : " << decimal;

break;

}

else {

cout << "이진수(1또는 0)만 입력 가능합니다!!!";

}

}

}

bool validateBinary(int num) {

string str = to\_string(num);//입력받은 정수를 문자열로 전환

for (int i = 0; i < str.size(); i++) {

if (str[i] > '1') {

return false;//10진수면 false

}

}

return true;//2진수면 true

}

int binaryToDecimal(int num) {

int sum = 0;

string str = to\_string(num);//입력받은 정수를 문자열로 전환

for (int i = str.size() - 1; i >= 0; i--) {

if (str[i] == '1') {

sum += 1 << (str.size() - 1 - i);//bit 연산을 통해 2진수 계산

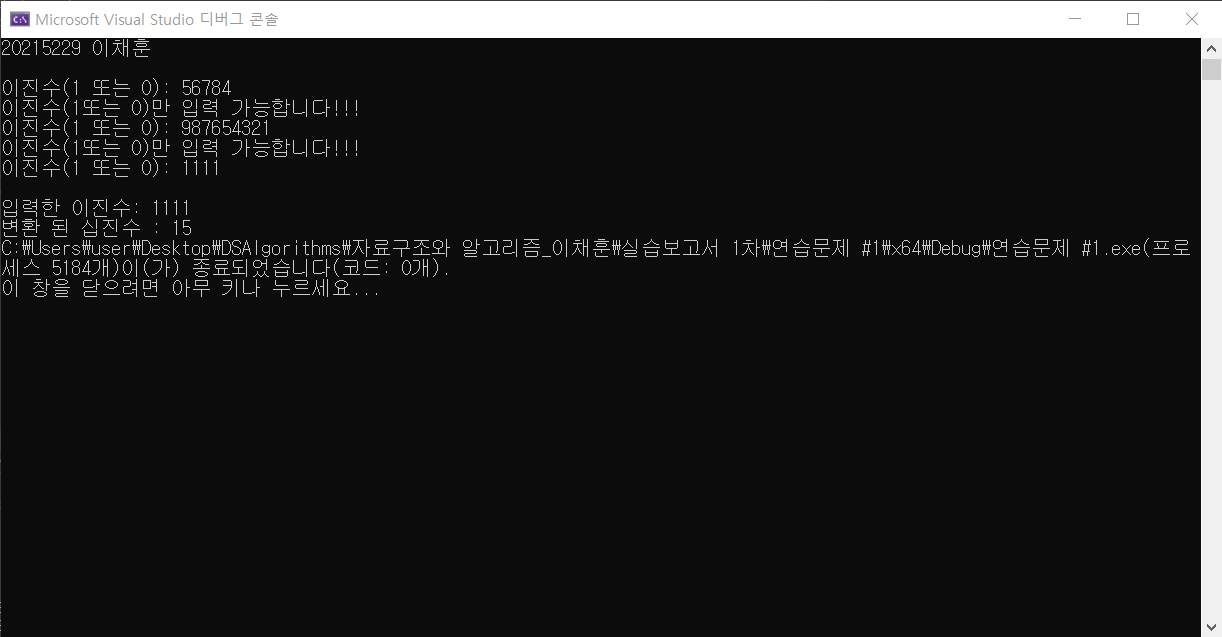
}

}

return sum;

}

**실행결과**

****

**연구조사**

**알고리즘**

1. num값 입력받기

2. validateBinary(int num)

2-1 입력 받은 정수를 to\_string을 이용하여 문자열로 변환

2-2 for문을 사용해, 각 index의 값이 1보다 큰지 확인

2-3 1보다 크면 false(10진수), 작으면 true(2진수)을 반환한다.

3-1. 반환값이 true일 경우, binaryToDecimal(int num) 실행

3-1-1 입력받은 정수를 to\_string을 이용하여 문자열로 변환

3-1-2 for문, if문, Bitwise operator

for문 : 문자열로 변환된 이진수의 오른쪽에서 첫째 자리부터 접근

if문 : 자릿수가 1인 경우

bitwise operator : str.size()-1-i 만큼 왼쪽 시프트 연산자를 통해 이진수 계산

3-1-3 계산된 값을 sum에 누적 합산

3-2. 반환값이 false인 경우 while(true)이므로, 재입력을 받는다.

**\*\*주요 사용 기능\*\***

**string 라이브러리의 to\_string**

to\_string(num) : 숫자형을 string 타입으로 변환

bitwise left shift(<<) : 2진수 계산을 비트 단위 연산자를 통해서 구현함.

**2. 데이터 정렬: 정적 배열과 동적 배열**

**소스코드**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int arr[10];

vector<int> v;

void inputValues() {

cout << "임의의 정수 10개를 입력하세요..." << endl;

for (int i = 0; i < 10; i++) {

cout << "arr[" << i << "] : ";

cin >> arr[i];

v.push\_back(arr[i]);

}

}//값 입력 및 정적, 동적 배열에 값 추가

int main(void) {

cout << "20215229 이채훈" << endl;

cout << endl;

inputValues();

sort(arr, arr + 10);//정적 배열 오름차순 정렬

cout << "정렬(정적 배열) : ";

for (int i = 0; i < 10; i++) {

cout << arr[i] << " ";

}//정적 배열 출력

cout << endl;

sort(v.begin(), v.end());//동적 배열 오름차순 정렬

cout << "정렬(동적 배열) : ";

for (int i = 0; i < v.size(); i++) {

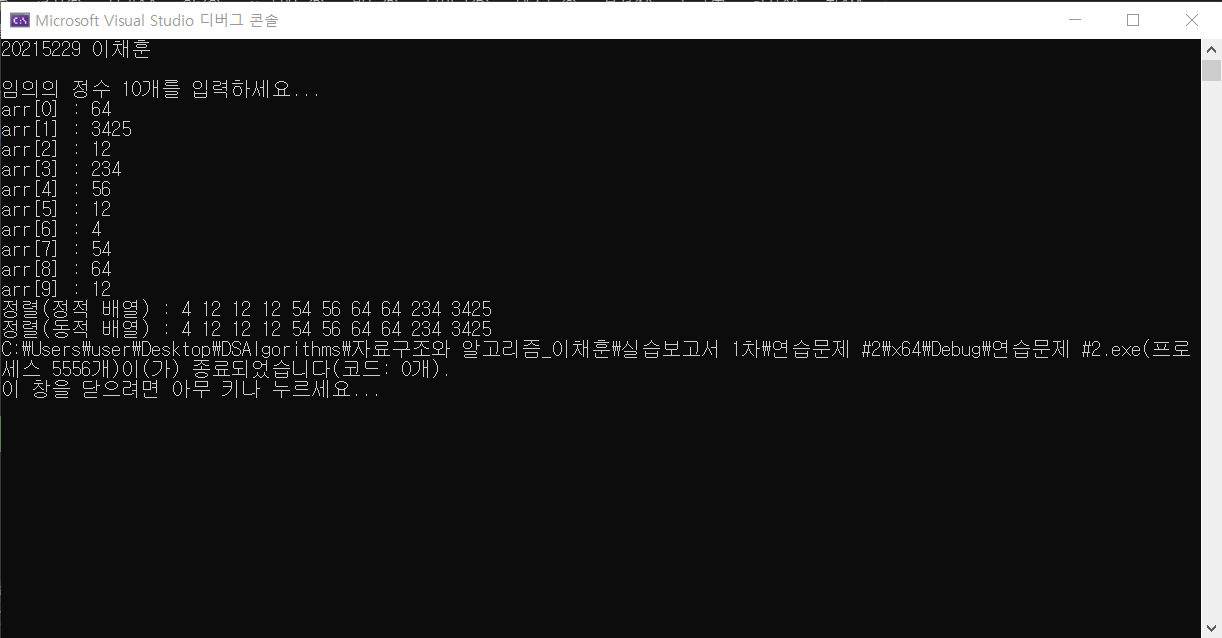
cout << v[i] << " ";

}//동적 배열 출력

return 0;

}

**실행결과**

****

**연구조사**

**알고리즘**

**\*\*주요 사용 기능\*\***

**vector 라이브러리**

**3. 학생 성적 처리: 동적 메모리 할당과 포인터 참조**

**소스코드**

#include <iostream>

using namespace std;

void stuGrade(const double\* pArr, int num, double& rSum, double& rAve, double& rMax);

/\*

세 개의 참조 매개변수(rSum, rAve, rMax)를 통해 결과를 반환함.

pArr이라는 const double\* 타입의 포인터, num이라는 int 매개변수를 입력으로 받음.

\*/

int main(void) {

int stuSize;

double\* pArr = nullptr;//동적배열의 포인터 초기화

double sum, ave, max;

cout << "20215229 이채훈" << endl;

cout << endl;

cout << "입력할 학생(성적) 수: ";

cin >> stuSize;

pArr = new double[stuSize];//입력 받은 학생 수에 따라, 동적으로 배열 할당

for (int i = 0; i < stuSize; i++){

cin >> \*(pArr + i);

}

stuGrade(pArr, stuSize, sum, ave, max);

cout << "\n ##### 성적 결과 출력 ##### \n" << endl;

cout << "학생수: " << stuSize << endl;

cout << "총 점: " << sum << endl;

cout << "평 균: " << ave << endl;

cout << "최대값: " << max << endl;

delete[] pArr; //동적으로 할당된 메모리 해체

return 0;

}

void stuGrade(const double\* pArr, int num, double& rSum, double& rAve, double& rMax) {

rSum = 0;

rMax = \*pArr;

for (int i = 0; i < num; i++) {

rSum += \*(pArr + i);//학생들의 성적 총점

if (rMax < \*(pArr + i)) {

rMax = \*(pArr + i);

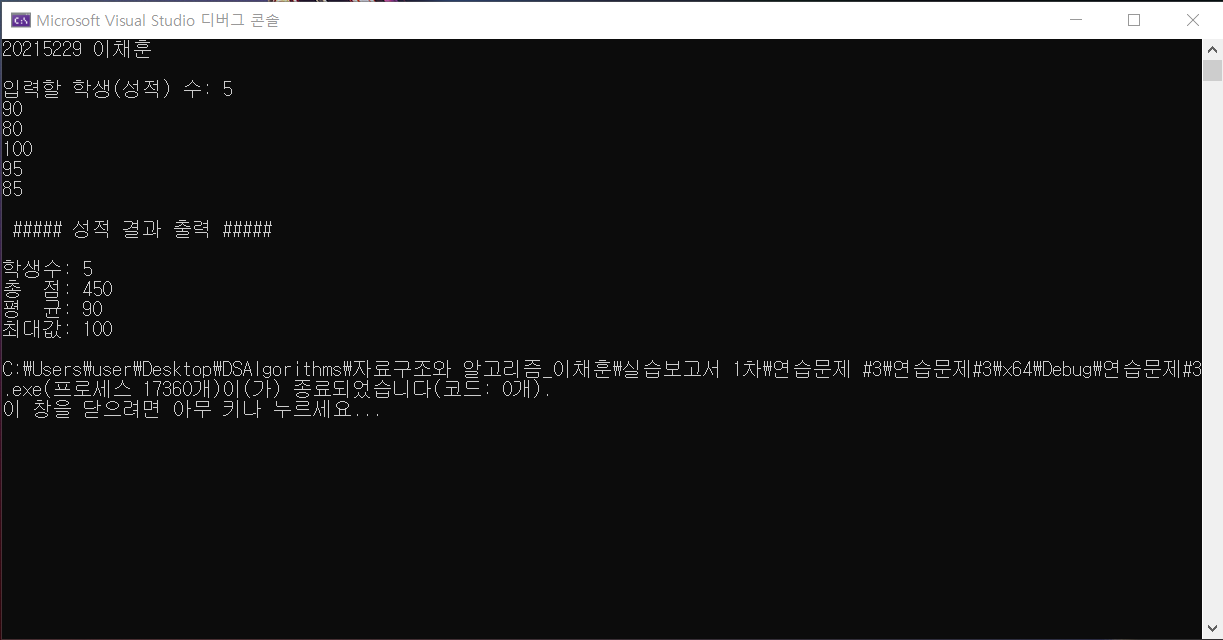
}//학생들중 성적 최대값

rAve = rSum / num;//학생들의 성적 평균

}

}

**실행결과**

****

**연구조사**

**알고리즘**

**\*\*주요 사용 기능\*\***

**동적 메모리 할당&배열의 포인터**

**4. 로또 복권 : 동적 메모리 할당과 2차원 배열 자동 생성**

**소스코드**

#include <iostream>

#include <random>//난수 생성을 위한 라이브러리

using namespace std;

void HappyLotto(int num);

int main (void) {

int num;

cout << "20215229 이채훈" << endl;

cout << endl;

cout << "로또 복권을 몇 장 구입 하실래요? "; cin >> num;

cout << endl;

cout << " ### 생성된 로또 번호 ### " << endl;

cout << endl;

HappyLotto(num);

return 0;

}

void HappyLotto(int num) {

const int NUM\_MIN = 1;

const int NUM\_MAX = 45;

//로또 번호 범위 1~45

random\_device rd;

//난수 생성기 초기화

mt19937 gen((rd()));

//메르센 트위스터 엔진 초기화

uniform\_int\_distribution <int> distribution(NUM\_MIN, NUM\_MAX);

//로또번호 1~45 난수 범위 설정

for (int i = 0; i < num; i++) {

int\*\* lotto = new int\* [6];

for (int j = 0; j < 6; j++) {

lotto[j] = new int;

}

//6개의 int형 포인터 원소를 갖는 2차원 동적 배열 생성

for (int j = 0; j < 6; j++) {

bool redundant;

do {

redundant = false;

\*lotto[j] = distribution(gen);

for (int k = 0; k < j; k++) {

if (\*lotto[k] == \*lotto[j]) {

redundant = true;

break;

}

}

} while (redundant);

}

/\*중복된 번호를 피하면서 로또 번호 생성.

일단 로또 번호를 생성 시키기 위해 do-while문 사용함.\*/

for (int j = 0; j < 6; j++) {

cout.width(3);

cout << \*lotto[j] << " ";

} cout << endl;

//생성된 로또 번호 출력, 간격3 맞추기

for (int j = 0; j < 6; j++) {

delete lotto[j];

}

delete[] lotto;

//동적 메모리 해체

}

}

**실행결과**

**텍스트, 스크린샷, 디스플레이, 컴퓨터이(가) 표시된 사진

자동 생성된 설명**

**연구조사**

**\*\*주요 사용 기능\*\***

2차원 배열, 동적 메모리 할당&해제

**난수 생성 방법(random/cstlib, ctime)**

**5. 문자열을 숫자(정수)로 변환 : atoi, stoi**

**소스코드**

#include <iostream>

#include <string>

using namespace std;

int my\_stoi(const string& rStr);

int main(void) {

string str;

cout << "20215229 이채훈" << endl;

cout << endl;

cout << "정수로 변환할 문자열 입력: "; cin >> str;

int result = my\_stoi(str);

cout << "입력 문자열: " << str << ", " << "변환된 숫자(int): " << result;

}

int my\_stoi(const string& rStr) {

const char\* ptr = rStr.c\_str();//문자열의 첫번째 문자에 대한 포인터

int result = 0;

int sign = 1;

if (\*ptr == '-') {

sign = -1;

++ptr;

}

else if (\*ptr == '+') {

++ptr;

}

while (\*ptr != NULL) {

if (\*ptr >= '0' && \*ptr <= '9') {

result = result \* 10 + (\*ptr - '0');

}

else {

break;

}

++ptr;

}

return result \* sign;

}

**실행결과**

**연구조사**

**알고리즘**

**\*\*주요 사용 기능\*\***

**1. stoi**

#include <string>

#include <iostream>

using namespace std;

int main (void){

    string s="123456789";

    cout<<"string s value : "<<s<<endl;

    int stoint = stoi(s);

    cout << "converted s value(INT) : "<<stoint<<endl;

    return 0;

}

/\*

실행결과

string s value : 123456789

converted s value(INT) : 123456789

\*/

/\*

stoid는 선행 공백 문자, +/- 기호, 16진수 접두사(0x또는0X)

, 여러 개의 0을 처리하고 정수를 올바르게 반환 할 수 있음.

숫자 앞의 다른 문자는 처리할수 없으며, 하나를 찾으면,

std::invalid\_argumetn 예외가 발생함.

\*/

**2. atoi**

: convert string to integer

atol : Convert string to long integer

atof : Convert string to double

strtol : Convert string to long integer

**6. 은행 계좌의 인증과 출금 : 응용**

**소스코드**

**실행결과**

**연구조사**

**7. Point 클래스 : 객체간 덧셈과 뺄셈 연산(멤버함수, 일반 함수)**

**소스코드**

**실행결과**

**연구조사**

**8. Person과 Student 클래스**

**소스코드**

**실행결과**

**연구조사**

**9. Albamon 클래스**

**소스코드**

**실행결과**

**연구조사**

**10. 다양한 자료형 출력 : 함수 템플릿**

**소스코드**

#include <iostream>

using namespace std;

template<class T>

void printArr(const T\* pArr, const int num) {

for (int i = 0; i < num; i++) {

cout << \*(pArr + i) << " ";

}

cout << endl;

}

char str[] = "Hi ~ Clickseo";

int iarr[] = { 10, 20 , 30, 40, 50 };

double darr[] = { 10.5, 20.5, 30.5, 40.5, 50.5 };

int main(void) {

cout << "20215229 이채훈" << endl;

cout << endl;

printArr(str, sizeof(str) / sizeof(char));

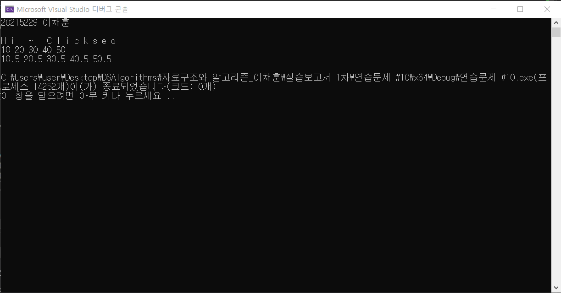
printArr(iarr, sizeof(iarr) / sizeof(int));

printArr(darr, sizeof(darr) / sizeof(double));

return 0;

}

**실행결과**

****

**연구조사**