

4, 5, 6 주차 과제 보고서

2024402055 박현지

No.1 문자열 정수 뽑아 정수로 출력하기

1. 문제 설명

사용자에게 문자열 입력 받고, 그 중 정수 뽑아 정수형으로 출력하기

2. 변수 설명

n: 문자열 길이

input 배열: 입력 받은 문자열

3. 프로그램 설명

while 문으로 무한 반복

문자열 길이(n) 받고 malloc 으로 동적할당

문자 입력(input) 받고, 0 만 있을 경우 무한 반복 종료

문자열 길이만큼 반복

아스키코드 이용하여 정수 문자일 경우 정수형으로 변환하여 출력

4. 실행 결과

```
길이를 입력하세요 : 10
정수 하나를 입력하세요 : 12345abc
입력한 정수는 :12345

길이를 입력하세요 : 0
정수 하나를 입력하세요 : 0
```

No.2 병합 정렬 구현

1. 문제 설명

20 개의 난수를 생성하고 (-100 ~ 100), 병합정렬 알고리즘으로 정렬

2. 변수 설명

merge_sort 함수: 배열 쪼개는 함수

merge 함수: 병합정렬 함수

arr 배열: 결과 숫자 배열

sort 배열: 정렬된 배열 임시 저장

mid, m: 배열 쪼개기 중간 값 (뒤 배열의 첫 번째 값)

left, l: 배열의 첫 번째 값

right, r: 배열의 마지막 값

3. 프로그램 설명

for 문 난수 생성, 저장, 출력

merge_sort 함수 호출

배열 크기가 1 이 될 때까지 하나의 배열 2 개로 쪼개기 (앞, 뒤)

merge 함수 호출

쪼개진 앞, 뒤 배열 첫 번째 값을 비교하여 정렬하기(sort)

정렬된 값(sort) 전체 배열로 옮기기 (arr)

결과 출력

4. 실행 결과

```
정렬 전 배열 :  
-41 -92 7 100 81 -33 -9 70 -85 74 51 21 72 -22 -17 81 85 36 48 42  
정렬 후 배열 :  
-92 -85 -41 -33 -22 -17 -9 7 21 36 42 48 51 70 72 74 81 81 85 100
```

No.3 M by N 행렬 A 와 AT 의 행렬곱

1. 문제 설명

M by N 행렬 난수(-10~10) 저장하고, $AT \cdot A$, $A \cdot AT$ 출력

2. 변수 설명

multiple 함수: 한 행, 한 열 곱한 값 반환

sum: 각 요소의 곱의 합

a 배열: m by n 초기 배열

at 배열: A transpose ($=AT$), n by m 배열

aat 배열: $A \cdot AT$ 배열

ata 배열: $AT \cdot A$ 배열

m,n: A 배열의 행, 열

3. 프로그램 설명

m, n 입력 받고 A, AT, $A \cdot AT$, $AT \cdot A$ 배열 동적할당

A 배열에 난수 생성, 저장

AT 배열 저장

A, AT 배열 출력

for 문 AAT 배열 호출

multiple 함수 호출

호출된 요소의 행, 열 값 곱하고 더해서 반환

AAT 배열 저장, 출력

for 문 ATA 배열 호출

mutiple 함수 호출

호출된 요소의 행, 열 값 곱하고 더해서 반환

ATA 배열 저장, 출력

4. 실행 결과

```
m by n 입력 (m n): 5 4
A 배열
 1 -8 -2 5
-8 6 -9 6
 3 0 -3 -10
 5 4 3 -7
-2 7 -4 -2

AT 배열
 1 -8 3 5 -2
-8 6 0 4 7
-2 -9 -3 3 -4
 5 6 -10 -7 -2
-----
A*AT 배열
 94 -8 -41 -68 -60
 -8 217 -57 -85 82
-41 -57 118 76 26
-68 -85 76 99 20
-60 82 26 20 73

AT*A 배열
103 -50 84 -104
-50 165 -54 -46
 84 -54 119 -47
-104 -46 -47 214
```

No.4 행렬식 구하기

1. 문제 설명

가우스 소거법 이용하여 N by N 행렬 A 의 행렬식 구하기

2. 변수 설명

gauss 함수: 주대각 성분 아래 값 0 만들기

a 배열: $A(n \text{ by } n)$ 행렬

tmp 배열: n 크기의 위치 교환용 일차원 배열

n : A 배열($n \text{ by } n$)의 행렬 크기

result: A 배열 행렬식

3. 프로그램 설명

크기(n) 입력 받고 $n \text{ by } n$ 크기 행렬 A 생성

크기 n 인 일차원 배열 tmp 생성

행렬 A 에 난수($-10 \sim 10$) 저장

A 행렬의 주대각 성분에 0 이 있을 시, 아래 행과 위치 변경하고 결과값에 -1 곱

gauss 함수 호출 ($n-1$ 회)

주대각 성분 아래 값 빼주기

결과값에 곱하기, 출력

4. 실행 결과

```
5
  9      2    -10      8      9
  7      3      5     -1     -8
 -1      2      0     -9     -5
 -9      7     -1      9      9
 -1      8      5     -4     -4
RESULT: -39031
```

No.5 로마숫자

1. 문제 설명

1000 이하의 숫자를 입력 받아 로마숫자로 변환

2. 변수 설명

num, num_save: 변환할 수

rom 배열: 로마숫자 입력 배열

num_count 배열: 각 자리별 숫자 계산

N: 자리수 (100, 10, 1)

rc: rom_count / rom 배열 차례로 쌓기

ten, five, one: 각 자리수 별 로마자 기호

3. 프로그램 설명

변환할 수 입력 받고 num_save 에 저장

초기 rom 배열 동적할당

for 문 100 자리수, 10 자리수, 1 자리수로 반복

단위수 별 10, 5, 1 나타내는 기호 설정

숫자 변환하기, rom 배열 확장하여 채워넣기

Rom 배열 마지막에 공문자 삽입하기, 결과 출력

4. 실행 결과

```
변환 수 입력 : 553
553=500+50+3=DLIII, 5
```

```
변환 수 입력 : 940
940=900+40+0=CMXL, 4
```

```
변환 수 입력 : 235
235=200+30+5=CCXXXV, 6
```

No.6 아다마르 행렬 구현

1. 문제 설명

$$H_{2^0} = [1], H_{2^1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, H_{2^2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, H_{2^n} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix}$$

2. 변수 설명

func 함수: 아다마르 행렬 구현

n: 아다마르 행렬 크기 지정(2^n by 2^n)

N: 2^n

mat 배열: 전체 아다마르 행렬 크기

3. 프로그램 설명

n 값 받아 $2^n(=N)$ 계산, 전체 행렬(mat) 동적할당

func 함수 호출

행렬의 크기가 1 이 될때까지 func 함수 호출

mat[N/2][N/2] 위치에는 -1 대입

결과 출력

4. 실행 결과

```
Enter the value of n for Hadamard matrix ( $2^n \times 2^n$ ): 3
Hadamard Matrix of size 8x8:
 1   1   1   1   1   1   1   1
 1  -1   1  -1   1  -1   1  -1
 1   1  -1  -1   1   1  -1  -1
 1  -1  -1   1   1  -1  -1   1
 1   1   1   1  -1  -1  -1  -1
 1  -1   1  -1  -1   1  -1   1
 1   1  -1  -1  -1  -1   1   1
 1  -1  -1   1  -1   1   1  -1
```