

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 전자전기 프로그래밍실습

과 제 명 : HW1

담당교수 : 민 형 복

학 과 : 전자전기공학부

학 년 : 1

학 번 : 2017311583

이 름 : 정해진

제 출 일 : 2018.3.26

1. Introduction

pointer의 올바른 사용법을 통해 특정 함수의 기능(예 : swap())을 온전히 발휘하게 한다. 1차원, 2차원 array를 사용할 수 있게 한다. 또한 malloc 함수를 통해 효율적인 메모리 사용을 꾀한다. 위의 3가지를 통해 사인, 코사인 함수를 출력한다.

2. Problem Statement

① Describe what is the problem.

- Main 함수에서 누락된 부분을 알맞게 채워 컴파일 및 실행되도록 한다.
- Main 함수를 채우는 과정에서 연습한 것으로 사인, 코사인 함수를 출력한다.

② Describe how do you solve the problem.

PART 1 : Array and pointer basics

[1] Pointers

[1-1] Pointer Basics

각각의 포인터들이 적절한 주소를 가리키도록 &를 통해 해결한다.

[1-2] Pointer for function output arguments

(A) num1, num2를 제외한 나머지 변수를 포인터를 통해 값을 처리한다.

(B) noSwap, doSwap 함수들의 형식은 강의자료 [1]을 참고한다.

[2] Array

[2-1] Get average of the above random numbers

aver_randnums 함수를 정의한다. 이 함수는 10개의 0부터 100 사이의 무작위 정수들을 받아 평균을 출력한다. 먼저 for문을 통해 Array에 존재하는 10개의 수들의 합을 sum_of_randnums에 저장하고, sum_of_randnums를 10으로 나눈 평균값을 return한다.

[2-2] Array as a pointer and dynamic memory allocation

(A) secondRandoms[0] = rand(); 를 사용할 수 없다. 왜냐하면 secondRandoms를 가리키는 pointer에 데이터를 저장할 memory가 부족하기 때문이다.

(B) secondRandoms에 malloc을 통해 메모리 할당을 하였으므로, [2-1]과 같은 방법으로 평균값을 리턴하는 함수를 정의하여 평균값을 출력한다. 메모리 할당 후엔 free 함수를 통해 반납한다.

[2-3] Array as a function argument

(A) array와 array의 길이를 input으로 주면 array의 모든 원소들을 출력하는 함수 printArray를 설계한다. for문을 통해 모든 원소들을 순서대로 출력해낸다.

(B) 0부터 10씩 증가하여 총 10개의 integer을 가지는 array를 만드는 함수 fillArray를 설계한다. for문을 통해 원소들을 순서대로 입력한다.

(C) 0부터 100씩 증가하여 총 10개의 integer을 가지는 array를 만드는 함수 fillArrayPointer를 설계한다. for문을 통해 원소들을 순서대로 입력한다. myNumbers에 malloc으로 메모리 할당을 먼저 시행한다.

[3] Strings

[3-1] String as a character array

(A) myname 이라는 이름을 가진 array에 초기 string값으로 교수님 성함을 입력한 후, "myname" array를 출력한다.

(B) myname 이라는 이름을 가진 array에 strcpy 함수를 통해 내 이름을 입력한 후, "myname" array를 출력한다.

[3-2] String as a pointer

(A) myname 이라는 이름을 가진 pointer가 가리키는 array에 초기 string값으로 교수님 성함을 입력한 후, "myname" 을 출력한다.

(B) method 1 : myname 이라는 이름을 가진 pointer가 가리키는 array에 내 이름이 적힌 string을 직접 입력하고 "myname" array를 출력한다.

method 2 : myname 이라는 이름을 가진 pointer가 가리키는 array에 malloc을 통해 메모리를 할당하고 strcpy 함수를 통해 내 이름을 입력한 후, "myname" array를 출력한다.

[3-3] String as a function argument

(A) myname 이라는 이름을 가진 pointer가 가리키는 array에 초기 string값으로 내 이름을 입력한 후, "myname" 을 출력한다.

(B) friend1name 이라는 이름을 가진 pointer가 가리키는 array에 내 친구 이름이 적힌 string을 직접 입력하고 "myname" array를 출력한다.

(C) friend2name 이라는 이름을 가진 pointer가 가리키는 array에 내 또 다른 친구 이름이 적힌 string을 직접 입력하고 더블 포인터를 이용해 "friend2name" array를 출력한다.

PART 2 : draw graphs of sine and cosine waves

(A) sin 함수를 주어진 조건 하에서 그릴 수 있어야 한다. drawSineWave 함수를 통해 graph[XSIZE][70] 라는 array에 사인 그래프에 해당하는 부분은 1, 나머지는 0을 저장하고 drawGraphArray 함수를 통해 graph[XSIZE][70]에 저장되어있는 사인함수를 출력해야 한다.

(B) cos 함수를 주어진 조건 하에서 그릴 수 있어야 한다. drawCosineWave 함수를 통해 malloc을 통해 메모리를 할당받은 graph라는 array에 코사인 그래프에 해당하는 부분은 1, 나머지는 0을 저장하고 drawGraphPointer 함수를 통해 graph에 저장되어있는 코사인함수를 출력해야 한다. drawGraphPointer 함수에서 유의해야 할 점은 (A)의 drawGraphArray 함수와 다르게 포인터를 통해 graph를 지정해야 한다. 그리고 malloc을 통해 할당받은 메모리를 다시 돌려줘야 하므로 disposeMemory 함수를 통해 할당받은 모든 메모리를 free 해야 한다.

③ Draw a flowchart of your algorithm

PART 1의 해결방법은 위 항목에서 제시하였으므로, PART 2의 해결방식을 보다 자세히 설명한다.

(A) main 함수에서 실행되는 함수는 drawSineWave 와 drawGraphArray 이다.

1. drawSineWave

이 함수는 char 자료형의 graph[][70], int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 sin함수 값에 해당하는 부분은 "1", 나머지는 "0"으로 저장되어야 하는 것이다. 이를 간단히 나타내면 다음과 같다.

```
graph[0] -----> 000000000000000000000000000000 ... 00000000000000000000
graph[1] -----> 111100000000000000000000000000 ... 00000000000000000000
graph[2] -----> 111111100000000000000000000000 ... 00000000000000000000
.
.
graph[XSIZE] -----> 111111111111111111111111111111 ... 111111111111111111
```

이런 식으로 "graph" array에 저장되어야 한다.

2. drawGraphArray

이 함수는 char 자료형의 graph[][70], int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 sin함수 값에 해당하는 부분("1"로 채워져 있는 부분)을

"+"로 전환하여 출력된 그림이다. 이를 수행하기 위해선 graph의 각 원소들을 "1"과 비교한 후, 참이면 "+"로 전환하여 출력한다.

(B) main 함수에서 실행되는 함수는 drawCosineWave 와 drawGraphPointer, disposeMemory 이다.

1. drawCosineWave

이 함수는 char 자료형의 graph를 가리키는 포인터, int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 cos 함수 값에 해당하는 부분은 "1", 나머지는 "0"으로 저장되어야 하는 것이다. (A)의 drawSineWave 함수와 유사한 방식을 따른다. 그 방식을 따라 "graph" array에 저장되어야 한다.

2. drawGraphPointer

이 함수는 char 자료형의 graph를 가리키는 포인터, int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 cos함수 값에 해당하는 부분("1"로 채워져 있는 부분)을 "+"로 전환하여 출력된 그림이다. 이를 수행하기 위해선 graph의 각 원소들을 "1"과 비교한 후, 참이면 "+"로 전환하여 출력한다.

3. disposeMemory

main 함수에서 malloc을 통해 graph의 메모리 할당을 했다. 그러므로 free 함수를 통해 할당받은 메모리를 전부 반납하는 기능을 해야 한다.

3. Implementation

PART 2의 해결방식을 변수 설명과 함께 작성한다.

(A) main 함수에서 실행되는 함수는 drawSineWave 와 drawGraphArray 이다.

1. drawSineWave

이 함수는 char 자료형의 graph[][70], int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 sin함수 값에 해당하는 부분은 "1", 나머지는 "0"으로 저장되어야 하는 것이다. 이를 간단히 나타내면 다음과 같다.

```
graph[0] -----> 000000000000000000000000000000 ... 00000000000000000000
graph[1] -----> 111100000000000000000000000000 ... 00000000000000000000
graph[2] -----> 111111100000000000000000000000 ... 00000000000000000000
.
.
graph[XSIZE] -----> 1111111111111111111111111111 ... 111111111111111111
```

이런 식으로 "graph" array에 저장되어야 한다.

이 과정을 실행하기 위해, double형 변수 num_y_sinValue와 double형 변수 degreeToRadian을 선언한다. num_y_sinValue는 sin값을 저장하기 위한 변수이다. degreeToRadian은 육십분법을 따르는 수치를 라디안 값으로 변환하고, 그를 저장하기 위한 변수이다. 이가 필요한 이유는 헤더파일 math.h에 존재하는 함수 sin()을 사용하려면 라디안 값을 넣어야 하기 때문이다.

for반복문을 통해 graph[i][j]에 "1"이나 "0"을 입력한다. i의 범위는 0부터 num_x_samples - 1 이다. i변수에 의존하는 반복문을 통해 0도, 3도, 6도 등의 각도에 해당하는 sin값을 저장할 특정 열의 array를 지정한다. j변수에 의존하는 반복문은 총 2개인데, 선행되는 반복문이 특정 열 graph[i]의 모든 원소를 "0"으로 채우는 과정을 수행한다. 그 이후로 실행되는 반복문이 그 각도에 해당하는 sin값까지 원소에 담긴 데이터를 "1"로 바꾸어 채우는 과정을 수행한다.

2. drawGraphArray

이 함수는 char 자료형의 graph[][70], int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 sin함수 값에 해당하는 부분("1"로 채워져 있는 부분)을 "+"로 전환하여 출력된 그림이다. 이를 수행하기 위해선 graph의 각 원소들을 "1"과 비교한 후, 참이면 "+"로 전환하여 출력한다. 비교를 하기 위해 int 자료형 변수 compare_number = "1"을 선언한다. 그리고 graph array의 각 원소들과 비교를 진행하기 위해 for 구문을 2번 중첩하여 특정 원소 graph[i][j]와 compare_number를 strcmp 함수를 통해 비교하고, 그 결과값을 받아 if 구문이 실행된다. 만약 그 결과값이 1, 즉 참이라면 특정 원소 graph[i][j]의 데이터를 "+"로 바꾸고, 이를 출력한다. 결과값이 0, 즉 거짓이라면 아무것도 출력하지 않는다.

(B) main 함수에서 실행되는 함수는 drawCosineWave 와 drawGraphPointer, disposeMemory 이다.

1. drawCosineWave

이 함수는 char 자료형의 graph를 가리키는 포인터, int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 cos 함수 값에 해당하는 부분은 "1", 나머지는 "0"으로 저장되어야 하는 것이다. (A)의 drawSineWave 함수와 유사한 방식을 따른다. 그 방식을 따라 "graph" array에 저장되어야 한다. 이 과정을 실행하기 위해, double형 변수 num_y_cosValue와 double형 변수 degreeToRadian을 선언한다. num_y_cosValue는 cos값을 저장하기 위한 변수이다. degreeToRadian은 육십분법을 따르는 수치를 라디안 값으로 변환하고, 그를 저장하기 위한 변수이다. 이가 필요한 이유는 헤더파일 math.h에 존재하는 함수 cos()을 사용하려면 라디안 값을 넣어야 하기 때문이다.

for반복문을 통해 graph[i][j]에 "1"이나 "0"을 입력한다. i의 범위는 0부터 num_x_samples - 1 이다. i변수에 의존하는 반복문을 통해 0도, 3도, 6도 등의 각도에 해당하는 cos값을 저장할 특정 열의 array를 지정한다. j변수에 의존하는 반복문은 총 2개인데, 선행되는 반복문이 특정 열 graph[i]의 모든 원소를 "0"으로 채우는 과정을 수행한다. 그 이후로 실행되는 반복문이 그 각도에 해당하는 sin값까지 원소에 담긴 데이터를 "1"로 바꾸어 채우는 과정을 수행한다.

2. drawGraphPointer

이 함수는 char 자료형의 graph를 가리키는 포인터, int 자료형의 num_x_samples와 num_y_samples를 input으로 받아 실행된다. 이 함수를 통해 얻어져야 하는 결과는 "graph" array에 cos함수 값에 해당하는 부분("1"로 채워져 있는 부분)을 "+"로 전환하여 출력된 그림이다. 이를 수행하기 위해선 graph의 각 원소들을 "1"과 비교한 후, 참이면 "+"로 전환하여 출력한다. 비교를 하기 위해 int 자료형 변수 compare_number = "1"을 선언한다. 그리고 graph array의 각 원소들과 비교를 진행하기 위해 for 구문을 2번 중첩하여 특정 원소 graph[i][j]와 compare_number를 strcmp 함수를 통해 비교하고, 그 결과값을 받아 if 구문이 실행된다. 만약 그 결과값이 1, 즉 참이라면 특정 원소 graph[i][j]의 데이터를 "+"로 바꾸고, 이를 출력한다. 결과값이 0, 즉 거짓이라면 아무것도 출력하지 않는다.

3. disposeMemory

main 함수에서 malloc을 통해 graph의 메모리 할당을 했다. malloc으로 graph array의 모든 행과 열에 대해 메모리를 할당했다. 그러므로 free 함수를 통해 행과 열에 할당받은 메모리를 전부 반납하는 기능을 해야 한다. 열에 할당받은 메모리를 free 하기 위해 for 구문을 사용했다.

4. Result

"2017311583.JungHaeJin.HW1.c" 를 컴파일 후 실행한 결과이다.

```
| Practice Pointers |
+-----+
## expecting the same 3 integers 1200.
miles = 1200 1200 1200
## expecting two numbers and add, sub of the numbers.
num1 = 20, num2 = 10
added = 30, subtracted = 10
## expecting no-swap, i.e. 20, 10.
after noSwap : num1 = 20, num2 = 10
## expecting do-swap, i.e. 10, 20.
after doSwap : num1 = 10, num2 = 20
```

PART 1 – [1] 실행 결과이다.

miles 출력 시 포인터와 더블 포인터가 가리키는 값 모두 잘 출력되었다.

num1, num2가 input으로 들어갔을 때 addsubtract 함수, noSwap, doSwap 함수가 제대로 실행되었다.

```
| Practice Arrays |
+-----+
## processing the following 10 numbers in array:
41 67 34 0 69 24 78 58 62 64
average of the above 10 integers = 49.70
## processing the following 10 numbers by dynamic alloc:
5 45 81 27 61 91 95 42 27 36
average of the above 10 integers = 51.00
## printArray() prints 10 integers in array:
10 20 30 40 50 0 0 0 0 0
## fillArray() put 10 numbers {0, 10, 20, ...} into array :
0 10 20 30 40 50 60 70 80 90
## fillArrayPointer() put {0, 100, 200, ...} into array :
0 100 200 300 400 500 600 700 800 900
```

PART 1 – [2] 실행 결과이다.

"firstRandoms" array에 입력된 0부터 100 사이의 무작위 10개의 정수들을 출력하고, "firstRandoms" array를 input으로 하여 무작위 정수들의 평균값을 average_of_randnums 변수에 저장하여 output 하는 aver_randnums 함수가 잘 실행되었다.

마찬가지로 "secondRandoms" array에 입력된 0부터 100 사이의 무작위 10개의

정수들을 출력하고, "secondRandoms" array를 input으로 하여 무작위 정수들의 평균값을 average_of_randnums 변수에 저장하여 output 하는 aver_randnums 함수가 잘 실행되었다. "secondRandoms" array를 다루는 과정에서 포인터가 사용되어, malloc을 통해 memory를 할당하고 free로 메모리를 반납하는 과정이 선행되었다.

이후로는 크기 10을 가지는 int 자료형의 "myNumbers" array에 대하여, 초기값 { 10, 20, 30, 40, 50 } 을 입력한 후, "myNumbers" array 를 input으로 받으면 그것을 출력하는 printArray 함수가 잘 실행된 모습이다. 또한 fillArray 함수, fillArrayPointer 함수가 각각 "myNumbers" array와 int 자료형을 가지는 배열 길이 len을 input으로 받는다. 그 후 "myNumbers" array에 0부터 10씩 증가하는 10개의 정수들, 0부터 100씩 증가하는 10개의 정수들을 저장한다. 그리고 "myNumbers" array를 printArray 함수를 통해 잘 출력되는 과정이 이루어진 모습이다.

```
! Practice Strings !
### init array string by professor's name :
Professor's name is 'Min, Hyoung Bok'
### init array string by strcpy() with your name :
My name is 'Jung, Hae Jin'
### init pointer string with professor's name :
Professor's name is 'Min, Hyoung Bok'
### set pointer string with your name by assignment :
My name is 'Jung, Hae Jin'
### set pointer string with your best friend's name by strcpy :
Name of my best friend is 'Jo, Jun Hui'
### printName() prints pointer string :
Jung, Hae Jin
### fillNameArray() prints pointer string :
Name of my first friend is 'Jo, Jun Hui'
### fillNamePinter() prints pointer string :
Name of my second friend is 'Cheon, Yeong Jun'
```

PART 1 – [3] 실행 결과이다.

(A), (B), (C) 각 항목에서 주어진 조건에 따라 string을 잘 출력한 모습이다.

5. Conclusion & Evaluation

실습을 통해 pointer을 다루는 과정을 좀 더 이해하게 된 것 같다. pointer을 잘못 사용하면 run time error가 뜨는 경우가 잦아서 c파일을 작성하는데 어려움이 있었지만, PART 1에서 그 어려움을 겪었기 때문에 PART 2에서 sine 함수와 cosine 함수를 그래프를 그려내는 여러 함수들을 설계하는데 큰 어려움이 없었다.

6. 참고 문헌

[1] Min, H. B. and SKKU, “ArrayPointer.pdf”