
Brand TIG

창원과학고등학교 정보동아리 TIG
2015학년도 활동보고서

머리말

김주영
웹 개발 Head

요즘 그 범위가 커지고 있는 소프트웨어를 만드는 일은, 정보과학보다는 코딩이나 프로그래밍이라는 이름으로 더욱 많이 접하셨을 것입니다. 하지만 이러한 보도나 전달들은 프로그래밍 내지는 소프트웨어 엔지니어링이 무엇인지 그 본질을 전달하지 못하고, 일종의 트렌드처럼 전달하고 있어 아쉬움이 남습니다. 사실 1년간의 정보 수업을 들으면서, 여러분들이 역시 프로그래밍은 내가 할 일이 아니라는 생각만 가지게 되셨을까하는 부분도, 정보과학 지망생으로서 염려가 됩니다. 그래서 정보동아리는 프로그래밍이 우리 주변에 가까이 있고, 우리 모두가 공감할 수 있는 꼭 필요한 것을 만드는 일이라는 점을 알리고 싶었습니다.

이러한 맥락에서, 우리는 정보동아리 활동을 하면서 정보과학이란 무엇인지, 그리고 소프트웨어 개발은 어떻게 이루어지며 우리 주변에서 어떤 것을 바꿀 수 있는지에 관해 많은 생각을 했습니다. 아직 그 결론은 얻지 못했습니다. 하지만 알 수 있었던 것은 이것입니다. 프로그래밍은, 기계가 생각하는 방식을 정해주는 것입니다. 사람이 하나씩 더하며 계산을 하듯, 컴퓨터에게 하나씩 더하며 계산을 하도록 시키는 과정입니다. 그렇게 하나씩 생각을 더하고 컴퓨터에게 부탁하는 방식을 바꿀 뿐입니다. 복잡한 코드는 전부, 기계가 일하기에 가장 적합한 사고방식은 무엇일까 하는 생각의 흔적입니다.

우리는 여기에 다 적지 못할 정도로 많은 시도를 했습니다. 팀으로도 했고, 개인적으로도 했습니다. 웹사이트도 만들었고, 게임도, 프로그램도 만들었습니다. 하지만 배우는 과정에서의 이러한 시도들의 가치는 공유할 때 비로소 극대화된다고 생각합니다. 프로그래밍이 우리 주변의 일이라는 점을 알리고 싶어 정보동아리에서 활동했고, 같은 이유로 이렇게 1년간의 활동을 공유하는 글을 씁니다. 이만 줄이겠습니다.

목차

1. 머리말
2. 동아리 소개 및 활동 내용 설명
3. CSHians 개발 보고서
4. 기초 PHP 문법 소개
5. 자료구조 및 알고리즘 소개
6. 세미나 활동 보고

동아리 소개

1. 정식 명칭: 창원과학교등학교 정보동아리 TIG

2. 구성 인원

구분	1학년 Head	1학년 부회장	웹 개발 Head	C언어 Head	2학년 Head
이름	서빈	강승환	김주영	차정민	정영광
경험 분야	C / C++ PHP / 서버 이외 다수	C / C++	C / C++ HTML / PHP	C / C++	C / C++

3. 주요 활동 내용

초기	정보올림피아드 대비 학습
중기	C언어 세미나
후기	프로젝트 활동(CSHians 개발)

★ 부록 및 기타

- 목차에 따라 활동 내용 설명, CSHians 개발 보고서, XE를 통한 홈페이지 제작 방법 소개, 기초 PHP 문법 소개, 자료구조 및 알고리즘 소개가 이어집니다.
- 2016년부터 정보동아리 TIG의 명칭이 전산동아리 Key.East(키이스트)로 변경됩니다. 첫 번째로 발행되는 이 정보동아리 활동보고서가 TIG의 마지막 기록물이 될 것입니다.
- 본 활동보고서는 <http://cshians.net/circles/academic/key.east/seminars> 내지는 키이스트 홈페이지에서도 보실 수 있습니다.

★ 활동 내용 설명

1. 정보올림피아드 대비(1분기)

1년간 C언어 학습을 하면서 많은 자료를 공유했습니다. 특히 정보올림피아드 대비 기간에는 동아리 시간을 이용해서 서로 C언어 문법을 익혔고, 여러 가지 기술을 공유했습니다. 이후에는 코드업에서 여러 가지 문제를 풀었으며, 기출문제를 풀이하였습니다.

2. C언어 세미나(2분기)

C언어 세미나를 위해 각자 인터넷에서 여러 가지 자료구조와 알고리즘에 대해 찾아보고 공부했으며, 개인별로 학습지를 제작하여 공유하였습니다. 최종적으로는 자신이 조사한 내용과 관련된 문제의 풀이를 프레젠테이션으로 준비하여 세미나를 하였습니다. 이 과정에서 카이스트 영재교육원 C언어 과정을 수강하는 학생이 있어 자료를 공유해서 스택, 큐, 트리, 그래프 등을 같이 공부하기도 했습니다.

3. 재능나눔 활동(2분기)

재능나눔 활동으로는 초등학생들에게 스크래치를 이용하여 프로그래밍의 기본적인 논리를 설명하였습니다. 이를 위해 그 직전 주에 스크래치를 해 보고, 어려워할 것으로 생각되는 부분을 고려하여 설명을 준비하였습니다. 당일에는 모르는 부분에 대해서 최대한 설명을 해 주었습니다.

4. 프로젝트 활동(3분기)

먼저 기초적인 PHP, HTML, CSS의 구조를 익히고, 기존 웹사이트들의 소스를 원하는 대로 수정하면서 이를 익혔습니다. 그리고 기존에 제공되고 있는 커뮤니티 서비스들을 살펴보고, 오픈소스로 공개되어 있는 카이스트 학내 커뮤니티 아래의 소스코드를 살펴보았습니다. 최종적으로는 XE를 설치하고 커뮤니티 서비스를 개발하였습니다. 이 과정에서 로그인 페이지, 디자인 수정 등을 하였고, CSS와 HTML의 연동 방법, HTML의 각종 태그 사용에 관한 확실한 이해를 할 수 있었습니다.

CSHians 개발 보고서

서문

창원과학고등학교 정보동아리 TIG는 학생 간의 개방적 소통문화에 학생사회의 미래가 있다고 믿고, 학생 전용 커뮤니티인 CSHians를 운영하기로 하였습니다.



현재 창원과학고등학교 학생들이 사용하는 소통수단은 주로 카카오톡과 페이스북입니다. 하지만 두 가지 모두 실질적인 필요를 충족시키지 못하고 있습니다. 특히 수행평가 자료와 같은 첨부파일 전달이 매우 어렵고, 휘발성이 짙은 SNS의 특성상 과거에 했던 말이나 올렸던 글을 찾기 어렵습니다. 최근 들어 전자 파일 제작을 통한 수행평가가 증가하고 있어 파일의 신속하고 편리한 전달과 과거 자료의 쉬운 접근성의 중요성이 커져가고 있음에도 불구하고, 학생들의 소통수단은 이에 도움을 주지 못하고 있습니다.

창원과학고등학교 정보동아리 TIG는 우리가 가진 능력으로 이를 해결하기 위한 방법으로 2015년 1년간 많은 생각을 했습니다. 새로운 SNS나 파일 첨부 서비스 소개도 거론되었습니다. 하지만 결국 SNS의 문제를 해결하기 위해서는 SNS에서 빠져나와야 할 것입니다. 결국 우리는 그 동안의 경험을 응집하여, 새로운 학생 전용 커뮤니티인 CSHians를 만들어 운영하기로 하였습니다.

〈창원과학고등학교 커뮤니티 서비스 CSHians〉는 학생 간의 수평적 의사소통 문화에 도움이 될 것입니다. 사회가 필요로 하는 인재는 꾸준히 바뀌지만, 최근에는 다른 사람과 쉽게 소통하는 능력도 그 중요한 자질로 떠오르고 있습니다. 재학생 및 졸업생이 같은 공간에서 생각을 나누고 의견을 교류하며 서로를 돕는 것으로부터 창원과학고등학교 학생들이 더 좋은 인재가 되는 기틀을 마련할 수 있다고 생각합니다. 이와 같은 커뮤니티 시스템은 전국 어떤 과학고등학교에서도 제공하지 못하고 있는 것입니다. CSHians는 미래 시대에 최적의 인재가 되기 위해 나아가는 창원과학고등학교 학생들을 지원할 뿐만 아니라 장기적으로 전국의 모든 학교에 새로운 형태의 전자적 의사소통 플랫폼을 제시하여 학생사회에 공헌하고자 합니다.

본문

1. 서비스 소개

- CSHians는 창원과학고등학교의 커뮤니티 서비스입니다.
- CSHians는 정보 제공과 커뮤니티 활성화의 목적으로 운영되고 있습니다. 정보 제공의 목적으로는 대학의 학사포탈을 표방하여 여러 가지 API 등을 이용해 각종 유용한 정보를 제공하고 이용자들이 자료를 올릴 수 있도록 하고 있으며, 커뮤니티 활성화의 목적으로는 게시판 서비스를 제공하고 있습니다. 추후 제공하는 서비스의 범위를 확장할 예정입니다.
- CSHians는 다음의 도메인에서 제공되고 있습니다. <http://cshians.net>

2. 서비스 현황

- 2016년 2월 말 현재, CSHians는 1.0.0-dev4 버전으로 운영되고 있습니다.
- 최초의 정식 버전 공개일은 2016년 4월 중으로 예상하고 있습니다.
- CSHians는 PHP 기반 오픈소스 CMS인 XpressEngine 1.8.17 버전에 기반하여 운영되고 있으며, mysql DB를 사용하고 있습니다.
- CSHians는 웹 호스팅을 통하여 제공되고 있습니다. 도메인은 cshians.net입니다.
- 학교의 FTP망이 차단된 관계로, 모든 업데이트는 주말에만 가능합니다. 현재 매주 주말마다 정기점검을 통하여 기능을 개선하고 업데이트를 진행하고 있습니다.

3. 개발 단계

- CSHians는 2015년 5월부터 9월까지 기획 기간을 거쳐 10월부터 개발이 시작되었습니다. 초기에는 Wix 등의 공개된 홈페이지 빌더를 사용했으나, 기능이 제한적이고 높은 가격으로 인해 CMS를 도입하여 직접 개발하기로 하였습니다.
- XE를 사용하기로 결정한 것은 11월 초입니다. 로컬 PC에 DB를 설치하는 것이 어려우므로, 결정 즉시 호스팅을 시작하여 XE를 호스팅 서버에 설치했습니다.
- 현재는 필요한 모듈과 애드온을 설치하고, 기존 코드를 수정하여 메인 페이지의 슬라이더나 상단바, 모바일웹, 로그인 페이지 등을 구축하였습니다.
- 차후 웹메일서비스나 동아리별 페이지 등 다양한 기능을 제공할 예정입니다.

부록

XE에 대한 소개

홈페이지를 만들기 위해서는 많은 기술이 필요한 것으로 알려져 있습니다. 하지만 실제로는 비전문가들도 따라만 하면 쉽게 자신만의 홈페이지를 만들 수 있습니다. 우리는 Wix와 같이 디자인만 열심히 하면 쉽게 만들 수 있지만 자유로운 제작이 어려운 서비스들에 비해, 쉽고 간단하면서 자유롭게 개발할 수 있는 CMS를 중심으로 홈페이지를 만들어보려 합니다.

CMS(Content Management System)는 사용자 각 개개인이 만드는 콘텐츠를 관리하는 소프트웨어로, 웹 분야에서는 주로 홈페이지를 만들고 콘텐츠를 관리하는 데 필요한 핵심 프로그램을 가리킵니다. 현재 국내 CMS에는 XE와 그누보드가 있고, 이외에도 해외에는 워드프레스 등의 CMS가 있습니다. Wix와 같은 홈페이지 빌더도 CMS로 보는 경우가 있습니다.

우리가 사용할 XpressEngine은 고영수씨(zero)가 개발했던 오픈소스 CMS인 제로보드에 기반합니다. 제로보드는 개발 당시가 2007년이었던 점에서 소규모 비전문가 사이트 개발자들에게 매우 유용한 프로그램이었습니다. 현재는 제로보드가 개선되면서 XpressEngine이라는 새로운 프로그램으로 바뀌었고, 현재는 개발자 지원 프로젝트인 D2 Project의 일환으로 네이버에서 개발을 맡고 있습니다.

부록

XE를 사용한 쉬운 홈페이지 만들기 매뉴얼

블로그를 만들 때, html 소스를 이용해서 자유롭게 편집이 가능한 티스토리보다는 많은 기능이 이미 만들어져 있는 네이버나 다음 블로그가 편리합니다. 마찬가지로, 홈페이지를 만들 때도 처음부터 모든 것을 직접 설계하는 것보다는 CMS를 이용하는 것이 훨씬 편리합니다.

CMS를 이용한 홈페이지 제작은 호스팅 및 CMS 설치, 그리고 환경설정 및 마무리의 단계로 이루어집니다. 각 단계의 순서에 맞춰 설명하도록 하겠습니다.

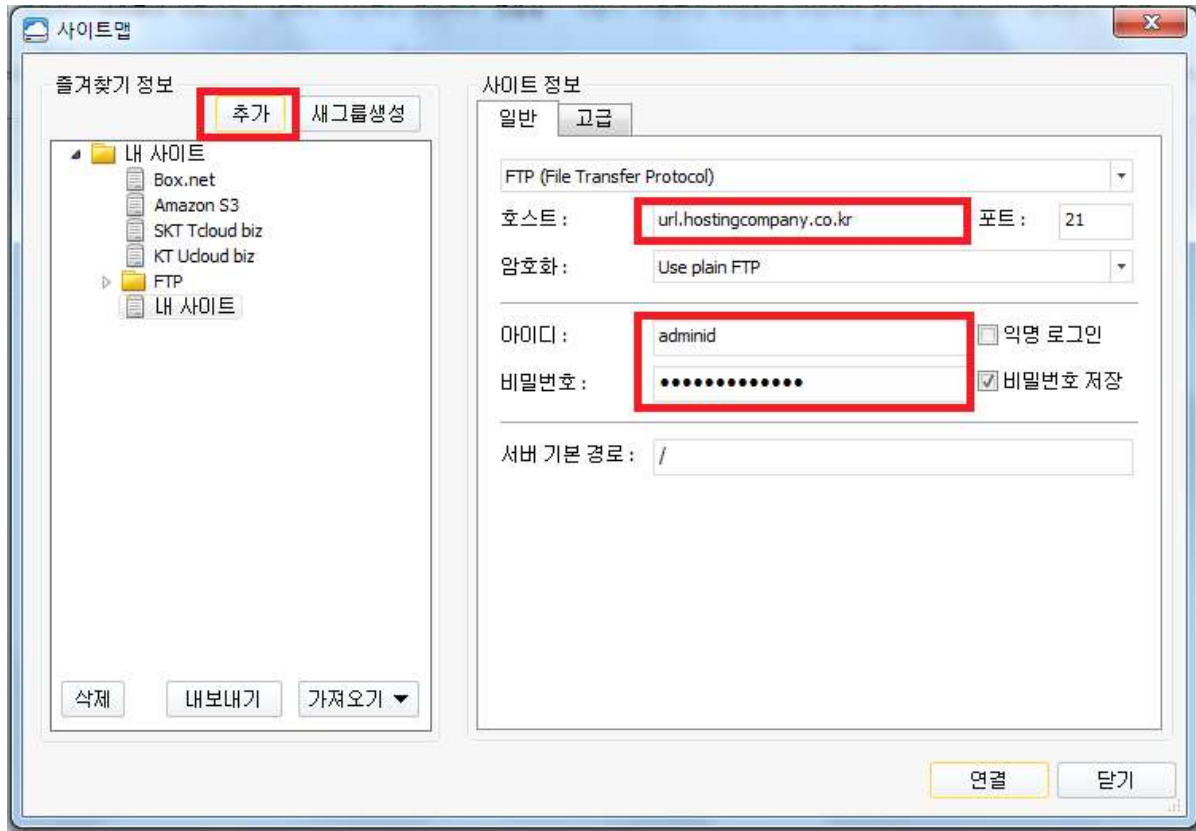
1. 호스팅 시작하기

본 매뉴얼에서는 비교적 저렴한 호스팅 업체들인 웹티즌(www.webtizen.co.kr)과 닷홈(www.dothome.co.kr)에 대해서 다루겠습니다.

- 호스팅 업체를 골라 리눅스호스팅을 신청합니다. 닷홈에서는 XE가 이미 설치된 호스팅을 제공하고 있습니다. 이것을 이용하면 설치 단계를 건너뛸 수 있지만, 직접 설치해보는 것을 권장합니다.
- 호스팅이 신청되었으면 해당 사이트를 방문해 봅니다. 보통 자신이 입력한 관리자 ID에 호스팅 업체의 도메인을 붙이면 됩니다. (예를 들면 `cshians.webtizen.co.kr` 내지는 `cshians.dothome.co.kr`)

2. 업로드하기

- XE 공식 사이트(www.xpressengine.com)를 방문하여 최신 버전의 XE를 다운받습니다. 사이트 우측 하단에 다운로드 버튼이 있습니다.
- 다운받은 XE를 호스팅하기 위해, FTP 클라이언트 프로그램을 설치합니다. 더욱 안정적인 파일질라를 설치해도 되지만, 초보자들에게는 좀 더 쉬운 알드라이브를 권장합니다. 알툴즈(www.altools.co.kr)에서 알드라이브를 다운받아 설치합니다.
- 알드라이브를 연 뒤, 연결 화면에서 필요한 정보를 입력하고 연결합니다.
- 알드라이브로 정상적인 접속이 되면 html 내지는 public_html 폴더가 보입니다. 더블클릭해서 그 폴더를 엽니다.
- 상단메뉴의 홈을 누르면 하위메뉴가 나오고 여거서 업로드를 클릭하여 업로드할 파일이 있는 폴더를 선택합니다. 루트 폴더인 html 폴더에 XE 파일을 업로드합니다.



3. 설치하기

- 자신이 호스팅한 사이트를 방문하여 XE 설치창의 안내에 따릅니다.
- 먼저 언어를 선택합니다.
- DB는 mysql로 선택합니다.
- DB 접속 정보는 다음과 같이 입력합니다. (cshians가 DB 아이디일 때)

	웹티즌	닷홈
DB 아이디	호스팅 계약시 설정한 대로 (cshians)	
DB 비밀번호	호스팅 계약시 설정한 대로	
DB 이름	cshians_db	cshians
DB 호스트네임	db2.webtizen.kr	localhost
DB Port	3306	
테이블 접두사	xe	

- 시간은 GMT+09:00으로 설정합니다.
- 관리자 정보에는 앞으로 사이트 관리를 위해 사용할 계정 정보를 입력합니다.

기초 PHP 문법 소개

PHP는 HTML과 CSS와 함께 3대 웹 언어 중 하나입니다. 최근에는 PHP를 멀리하는 경향이 있지만, PHP는 아직 대부분의 웹 사이트 개발에 사용되고 있습니다.

PHP의 기본 문법은 C와 비슷한데, PHP로 만들어진 코드는 다음과 같이 PHP 태그로 둘러싸여 있어야 합니다. 출력은 printf() 함수를 사용해도 되고, 아래와 같이 PHP의 echo 구문을 사용해도 됩니다.

```
key.east.php
1 <?php
2     printf("Hail Key.East");
3     echo "Glory Key.East";
4 ?>
```

PHP에서는 변수를 선언할 때 자료형을 말할 필요가 없고, 변수 앞에 \$을 붙여 구분합니다. 웹에서는 다양한 문자를 표시해야 하기 때문에 변수명을 텍스트로 사용할 경우가 많아 \$을 붙여 구분하는 것입니다.

```
key.east.php
1 <?php
2     $a = 50;
3     printf("%d", $a);
4 ?>
```

PHP의 가장 큰 특징은 오류를 무시하고 실행된다는 것입니다. C의 경우 컴파일을 할 때 어떤 오류가 있는지 알려주면서 작동을 거부합니다. 하지만 PHP는 그 부분을 무시해버립니다. 아래처럼 0으로 나누기를 하여 출력하라고 시키면 0이 출력됩니다.

```
key.east.php
1 <?php
2     $a = 10;
3     $b = 20 / 0;
4     printf("%d", $a); //10 출력
5     printf("%d", $b); //?? -> 0 출력
6 ?>
```

자료구조 및 알고리즘 소개

자료구조는 여러 가지 정보를 쌓아두는 방법이고, 알고리즘은 이 정보를 처리하는 방법이라고 볼 수 있습니다. 자료구조에는 스택, 큐, 트리, 그래프 등이 있고, 알고리즘에는 DFS, BFS, 동적계획법, 다익스트라 알고리즘 등이 있습니다.

자료구조는 왜 사용할까요? 여기서 보여주고 있는 자료구조의 소스는 사실 매우 불합리한 적용 사례입니다. 그냥 배열을 사용해도 상관없습니다. 하지만 실제로 사용되는 자료구조는 방대한 양의 자료를 매우 안정적으로 처리할 수 있습니다. 배열을 만드려면 RAM의 메모리 공간 일부를 배열에게 제공해야 하는데, 여기서는 그렇게 하지 않았지만 자료구조에서 내용물 하나를 꺼낸 다음 그 내용물을 지우고, 그 부분이 차지한 메모리를 다시 시스템에게 돌려주면, 컴퓨터가 더 많은 RAM 용량을 사용할 수 있게 됩니다. 하지만 그냥 배열에 저장한다면 프로그램이 꺼질 때까지 그 만큼의 메모리는 비어 있는 채로 예약되어 있어 낭비가 됩니다. 특히 게임을 만든다거나 할 때, 메모리를 낭비하는 경우 게임이 중간에 꺼지게 되어 자료구조를 잘 이용해야 합니다.

그럼 알고리즘은 왜 사용할까요? 우리가 1부터 1000까지의 수를 더하는 계산을 할 때, $n=1000$ 으로 본다면 우리는 덧셈 계산을 1000번, 즉 n 번 해야 합니다. 이와 같이 어떤 문제를 해결할 때, 자료의 크기가 n 일 때 몇 번의 계산을 해야 하는지를 시간복잡도라고 합니다. 이와 같은 경우에는 시간복잡도가 n 이므로, 1차의 시간복잡도를 가지고, $O(n)$ 으로 표현합니다. 알고리즘을 사용하면 문제를 풀 때 일일이 계산하는 것보다 이 시간복잡도를 매우 줄일 수 있습니다. 결과적으로 프로그램이 빨라집니다. 실제로, 어떤 알고리즘을 적용하는지에 따라 계산 속도가 몇 배씩 달라질 수 있습니다.

대표적인 것에 대해서 설명하고자 합니다. 여기서는 스택과 큐의 자료구조를 설명하고, 동적계획법의 알고리즘을 설명할 것입니다.

자료구조

1. 스택

스택은 배열을 기반으로 한 후입선출형 자료구조입니다. 즉, 늦게 들어온 자료가 먼저 빠져 나갑니다. 스택은 쉽게 생각하자면 일종의 통과 같습니다. 자료를 순서대로 넣으면, 꺼낼 때는 거꾸로 꺼내집니다. 거꾸로 꺼내기 위해서는 그냥 배열에 단어를 저장해서 이를 거꾸로 출력하면 되지 않냐고 생각할 수 있는데, 그것이 스택의 기본 원리입니다. 이해를 돕기 위해서, 들어온 단어를 뒤집어 출력하는 간단한 프로그램을 만들어 봅시다.

먼저 `gets()` 함수를 이용하여 문자를 임시 배열에 받아옵니다. 그리고 `push()` 함수를 정의하여 받아온 문자들을 처음부터 스택 배열에 하나씩 넣습니다. 마지막으로, `pop()` 함수를 정의하여 스택에서 문자들을 하나씩 꺼내옵니다. 주의할 점은, 스택에서는 문자를 하나 꺼냈으면 어디부터 꺼낼 수 있는지 다시 표시해야 한다는 점입니다. 만일 그렇게 하지 않는다면 같은 문자가 계속 꺼내집니다. 아래는 그 소스코드입니다.

```
key.east.cpp
1 #include<stdio.h>
2 #include<string.h> //strlen()을 사용하기 위한 헤더 파일
3
4 int data=-1;
5 char stack[30]; //여러 함수에서 쓰일 스택 관련 변수는 전역으로 선언
6
7 void push(char x)
8 {
9     data+=1; //data를 먼저 하나 올리고
10    stack[data]=x; //받은 문자를 스택에 넣습니다.
11 }
12
13 char pop()
14 {
15     data-=1; //data를 먼저 하나 내리고
16     return stack[data+1]; //들어있는 문자를 리턴합니다.
17 } //리턴을 이용하기 위해서 어쩔 수 없이 먼저 내리고 다시 data+1번째를 출력.
18
19 int main()
20 {
21     char arr[30]; //일단 문자를 받아둘 배열을 선언합니다.
22     int i, len;
23
24     gets(arr); //문자를 받아옵니다.
25     len=strlen(arr); //마지막 칸의 번호를 확인!
26
27     data=-1; //한편 data는 -1로 초기화해둡니다.
28
29     for (i=0; i<len; i++) {
30         push(arr[i]); //받아놓은 문자들을 push합니다.
31     }
32
33     for (i=len-1; i>=0; i--) { //i를 저렇게 한 것은 의미가 없습니다~
34         printf("%c", pop()); //스택에서 하나씩 꺼내서 출력합니다.
35     }
36
37     return 0;
38 }
```

2. 큐

큐는 배열을 기반으로 한 선입선출형 자료구조입니다. 즉, 먼저 들어온 자료가 먼저 빠져나갑니다. 큐는 쉽게 생각하자면 일종의 깔때기와 같습니다. 자료를 순서대로 넣으면, 꺼낼 때도 순서대로 나옵니다. 그대로 꺼내기 위해서는 그냥 배열에 숫자를 저장해서 이를 그대로 출력하면 되지 않냐고 생각할 수 있는데, 그것이 큐의 기본 원리입니다. 이해를 돕기 위해서, 큐를 이용해 빠진 숫자를 찾는 간단한 프로그램을 만들어 보시다. 1부터 10까지의 수 중 9개만 주어진 수열이 있습니다. 어떤 숫자가 빠졌는지 찾으려 합니다.

먼저 숫자를 하나씩 받아오면서 이것을 enqueue() 함수로 큐에 넣습니다. 그리고 숫자를 dequeue() 함수로 하나씩 빼내면서 있는 숫자는 배열의 해당 칸에 1을 표시해서 체크합니다. 그리고 배열의 칸에 1이 아닌 숫자가 있는 칸을 확인해서 빠진 숫자를 찾습니다. 아래는 그 소스코드입니다.

```
key.east.cpp x
1  #include<stdio.h>
2
3  int queue[20];
4  int rear=-1, front=-1;
5
6  void enqueue(int x) //일반적인 enqueue 함수입니다.
7  {
8      rear+=1;
9      queue[rear]=x;
10     return;
11 }
12 int dequeue() //일반적인 dequeue 함수입니다.
13 {
14     front+=1;
15     return queue[front];
16 }
17
18 int main()
19 {
20     int i, n, arr[20]={0};
21
22     for (i=0; i<9; i++) {
23         scanf("%d", &n);
24         enqueue(n);
25     } //숫자 9개를 큐에 넣습니다.
26
27     for (i=0; i<9; i++) {
28         n=dequeue();
29         arr[n]=1;
30     } //숫자를 빼내면서 배열에 체크합니다.
31
32     for (i=0; i<9; i++) {
33         if (!arr[i]) { //배열의 숫자가 0인 경우, 즉 배열 방 번호의 숫자가 없었다면
34             printf("%d", i); //해당 번호를 출력
35             return 0; //종료
36         }
37     }
38 }
```

알고리즘

동적계획법

동적계획법은 큰 문제를 여러 개의 작은 문제로 만들어 푸는 방법입니다. 간단한 예제를 접해 봅시다. 피보나치 수열을 구하는 함수는 보통 이렇게 작성합니다.

```
key.east.cpp
1 int fibo(int n)
2 {
3     if (n<=0) return 0;
4     else if (n==1) return 1;
5     else return fibo(n-2) + fibo(n-1);
6 }
7
```

이렇게 작성한 코드를 실행해 봅시다. $n=100$ 이 주어졌습니다. $\text{fibo}(100)$ 을 계산하기 위해서 컴퓨터는 $\text{fibo}(99)$ 와 $\text{fibo}(98)$ 을 계산하고, 이것을 계산하기 위해서 더 작은 숫자를 계산하면서 계속 더합니다. 그런데 이 과정에서 같은 값을 여러 번 계산하고 있습니다.

```
key.east.cpp
1 int fibo(int n)
2 {
3     if (n<=0) return 0;
4     else if (n==1) return 1;
5     else if (arr[n]>0) return arr[n];
6     else {
7         int temp;
8         temp=fibo(n-2)+fibo(n-1);
9         arr[n]=temp;
10        return temp;
11    }
12 }
```

이렇게 바꾸면, 매번 계산할 필요 없이 이전에 계산한 값이 있으면 배열에서 꺼내다 쓰기 때문에, 시간이 매우 단축됩니다.

그런데 이것은 단순히 시간을 줄이는 행위라고 볼 수도 있지만, 그렇지 않습니다. 이전에는 그냥 계산이 될 때까지 계산하는 코드였다면, 바뀐 뒤에는 작은 숫자를 계산해서 큰 숫자를 계산하겠다는 발상이 들어가 좀 더 가치 있는 코드가 되었습니다. 체계적으로 문제를 해결하는 것이지요. 이렇게 큰 숫자나 어려운 문제를 해결할 때, 작은 단계를 거쳐 해결하는 것이 동적계획법입니다.

동적계획법을 사용하는 문제 중 가장 대표적인 것은 LIS 문제입니다. 아무렇게나 주어진 수열 중에, 몇 개의 숫자를 뽑아서 그 순서를 바꾸지 않고 점점 커지는 수열을 만든다고 합니다. 이때 가장 길게 뽑을 수 있는 수열의 길이가 LIS입니다. 예를 들어서,

1 3 2 9 7 8 5 10

이런 수열에서 순서를 바꾸지 않고 뽑을 수 있는 가장 긴 수열의 길이는 5입니다. 임의의 수열이 주어질 때, LIS를 구하는 것이 LIS 문제입니다. (코드업 3735번 문제)

이 문제를 해결하기 위해서는 동적계획법을 사용해야 합니다. 먼저 수열이 n 개의 항으로 주어진다고 생각해봅시다. n 번째 항에서의 LIS는 n 번째 항이 LIS 수열에 포함된 경우의 LIS와, $n-1$ 번째 항까지의 LIS 중 하나입니다. 따라서 재귀적인 함수가 됩니다.

우리는 숫자를 하나씩 받아들이면서, 그 숫자까지 포함된 LIS를 계산하고, 이것이 그 전까지 계산한 LIS보다 큰지 확인할 것입니다. 이 과정을 n 번째까지 반복하면 최종적인 LIS가 계산됩니다. 아래는 이것을 코드로 구현한 것입니다.

```
key.east.cpp
1  #include<stdio.h>
2  int arr[1001], dp[1001];
3  int main()
4  {
5      int n, i, j, max, max_=0;
6      scanf("%d", &n);
7      dp[1]=1;
8
9      for (i=1; i<=n; i++) {
10         scanf("%d", &arr[i]);
11         max=0;
12         for (j=1; j<i; j++) {
13             if (max<dp[j]&&arr[j]<arr[i]) max=dp[j];
14         }
15         dp[i]=max+1;
16         if (max_<dp[i]) max_=dp[i];
17     }
18
19     printf("%d", max_);
20     return 0;
21 }
```


세미나 활동 보고

1년간 C언어 학습을 하면서 많은 자료를 공유했습니다. 동아리 시간을 이용해서 코드업에서 여러 가지 문제를 풀었고, 인터넷에서 여러 가지 알고리즘에 대해 찾아보고 공부했습니다. 카이스트 영재교육원 C언어 과정의 자료를 공유해서 스택, 큐, 트리, 그래프 등을 같이 공부하기도 했습니다.

이외에도 PHP와 HTML의 기초적인 문법을 익혔고, 이를 바탕으로 CSHians 웹사이트를 개발하였습니다. 개발 과정에서 코드를 공유하고 수정 방향을 서로 제시하였습니다.