



C Piscine

Rush 02

*Summary:* 이 문서는 42 C Piscine 과정 Rush 02 모듈의 학습 주제입니다.

# Contents

I	Instructions	2
II	Foreword	3
III	Subject	4
IV	Bonus	6

# Chapter I

## Instructions

- 그룹은 코렉션에 자동으로 등록됩니다. 등록은 두 번 할 수 없으므로 그룹 등록을 취소하지 마세요.
- 과제와 관한 의문은 과제를 더욱 복잡하게 만들 수 있습니다.
- 모든 과제물을 제출할 때는 제출 절차를 따라야 합니다.
- 이 문서는 제출 1시간 전까지도 변경될 수 있습니다.
- Moulinette은 -Wall -Wextra -Werror 플래그를 지정하여 컴파일하며 gcc를 사용합니다.
- 프로그램이 컴파일되지 않으면 0점을 받게 됩니다.
- Makefile을 제출해야 하며, 이 파일의 \$NAME, clean and fclean 규칙을 사용하여 프로젝트를 컴파일합니다.
- 할당된 팀원과 프로젝트를 함께 하고, 등록된 코렉션 시간에 모든 팀원과 함께 참석해야 합니다.
- 그룹의 모든 구성원은 프로젝트의 작업 내용에 대해 완벽하게 이해하고 있어야 합니다. 작업을 나누어 하기로 결정한 경우에도 모든 구성원이 서로의 작업 내용을 이해하고 있어야합니다. 코렉션 동안에 관련 질문을 받게 되며, 최종 점수는 코렉션 중 가장 설명을 못한 구성원을 기준으로 채점됩니다.
- 그룹에서 빠지는 사람이 없도록 쟁기는 것도 여러분의 책임입니다. 전화, 이메일, 비둘기, 텔레파시 등 그룹 구성원에서 연락할 수 있는 모든 수단을 사용하세요. 변명 해봤자 소용없습니다. 인생이란게 원래 늘 공정하지는 않으니까요.
- 하지만, 정말 온갖 방법을 다 동원해 봐도 연락이 안되는 그룹 구성원이 있다면, 어쨌든 프로젝트는 진행하세요. 그러면 저희가 코렉션 중에 그 점을 고려해 보도록 하겠습니다. 그룹 리더가 없는 경우에도 여러분은 제출 디렉토리에 접근할 수 있습니다.
- 굳이 말할 필요도 없겠지만, 과제는 Norm을 지켜야 합니다. 꼼꼼하게 작성하세요. 그럼, 재밌게 풀어 보세요!

# Chapter II

## Foreword

다음은 전통 방식의 피칸 파이 레시피입니다. :

재료:

페이스트리 반죽  
무염 버터 스틱  $\frac{3}{4}$   
황설탕  $1\frac{1}{4}$ 컵  
콘 시럽  $\frac{3}{4}$ 컵  
바닐라 익스트랙트 2 티스푼  
오렌지 껌질  $\frac{1}{2}$  티스푼  
소금  $\frac{1}{4}$  티스푼  
알이 큰 달걀 3알  
피칸 2컵 ( $\frac{1}{2}$  파운드)  
곁들임: 휘핑 크림 또는 바닐라 아이스크림

준비:

오븐 가운데 칸에 베이킹 시트를 깔고 화씨 350도로 예열합니다.  
밀대에 밀가루를 묻히고 바닥에도 밀가루를 살짝 뿌린 후, 밀대로 반죽을 밀어 지름 23 센티미터 정도의 파이 틀에 맞도록 지름 30센티미터 크기로 반죽을 펴칩니다.  
1.3 센티미터 정도 여유를 두고 반죽 가장자리를 잘라냅니다.  
튀어나온 부분을 아래로 접고 파이 틀 가장자리에 대고 살짝 눌러 주름 모양으로 자릅니다.  
반죽 바닥 전체를 포크로 찍어 구멍을 냅니다.  
반죽이 굳을 때까지 최소 30분 동안 식힙니다. (또는 10분 동안 냉장 보관)  
그동안 작은 소스팬에 버터를 넣고 중간불에서 녹입니다. 황설탕을 넣고 다 녹을 때까지 젓습니다.  
불을 끄고 옥수수 시럽, 바닐라, 오렌지 껌질, 소금을 넣고 저어 섞습니다.  
중간 크기의 오목한 그릇에 달걀을 넣고 살짝 푼 다음 옥수수 시럽 혼합물을 넣고 저어 섞습니다.  
만들어 놓은 파이지에 피칸을 넣고 그 위에 옥수수 시럽 혼합물을 평평하게 부어줍니다.  
뜨거운 베이킹 시트에 올려 놓고 50분~1시간 정도 필링이 굳을 때까지 굽습니다.  
완전히 식힙니다.

참고:

파이는 먹기 하루 전날 구워 냉장고에 보관해도 됩니다. 먹기 전에 실온에 꺼내 듭니다.

# Chapter III

## Subject

	Exercise 00
	rush-02
Turn-in directory :	<i>ex00/</i>
Files to turn in :	Makefile and all the necessary files
Allowed functions :	write, malloc, free, open, read, close

- 숫자를 인자로 취하여 그 숫자를 영어 단어로 변환하는 프로그램을 작성하세요.
- 실행 파일 이름: rush-02
- 여러분의 소스 코드는 다음과 같이 컴파일될 것입니다. :

```
make fclean  
make
```

- 여러분의 프로그램은 최대 2개의 인자를 가질 수 있습니다. :
  - 인자가 한 개인 경우, 그 인자를 변환해야합니다.
  - 인자가 두 개라면 첫 번째 인자는 새로 사용해야할 딕셔너리이고 두 번째 인자는 변환해야 하는 값입니다.
- 인자가 유효한 양의 정수가 아닌 경우 프로그램은 ‘Error’를 반환한 후 줄바꿈이 되어야 합니다.
- 프로그램은 프로젝트에 대한 자료로 주어진 딕셔너리를 파싱해야 합니다. 그 안의 값들이 결과를 출력하는 데 사용되어야 합니다. 이러한 값들은 수정될 수 있습니다.
- 힙 영역에 할당된 메모리(malloc(3) 사용)는 올바르게 해제되어야 합니다. 평가 중에 이를 확인할 것입니다.
- 딕셔너리는 다음 규칙을 가집니다.

```
[a number] [0 to n spaces]:[0 to n spaces] [any printable characters]\n
```

- 딕셔너리에 있는 값의 앞 뒤 공백은 잘라내야 합니다.
  - 주어진 기본 딕셔너리에 있는 모든 키(Key)는 새로운 딕셔너리에도 항상 존재합니다. 값(Value)는 수정될 수 있으며, 더 많은 엔트리가 추가될 수 있습니다. 하지만 초기에 주어진 키는 삭제할 수 없습니다.
  - 초기 엔트리만 사용해도 됩니다. (예를 들어 54: fifty-four를 추가했더라도, 50: fifty과 4: four를 사용해야 합니다.)
  - 딕셔너리의 엔트리는 정렬되어 있지 않을 수도 있습니다.
  - 딕셔너리에는 빈 행도 있을 수 있습니다.
  - 딕셔너리 파싱 과정에서 오류가 있을 경우 프로그램은 다음을 출력해야합니다: "Dict Error\n"
  - 딕셔너리에서 요청된 값을 해결할 수 없는 경우, 프로그램은 다음을 출력해야 합니다: "Dict Error\n".
- 예:
- ```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
$> grep "20" numbers.dict | cat -e
20 :      hey      everybody !$
$> ./rush-02 20 | cat -e
hey everybody !$
```

# Chapter IV

## Bonus

- 키값 데이터 사용자화하여 사용해보기: 50: fifty와 4: four 대신 54: fifty-four 사용하기
- 올바른 작성된 구문에 가깝도록 -, ,, and 사용하기
- 동일한 과제를 다른 언어로 해보기. 이를 위해 필요한 엔트리가 포함되는 다른 사전을 제공해도 됩니다.
- 인자가 없을 때 `read`를 사용하여 표준 엔트리를 읽기
- 본인이 생각하기에 프로젝트에 무언가를 더할 수 있는 그 밖의 기능