



C Piscine

Rush 00

Summary: 이 문서는 C Piscine@42의 Rush00 모듈용 과제입니다.

Version: 5.2

Contents

I	Instructions	2
II	Foreword	4
III	Main subject	5
IV	Rush 00	7
V	Rush 01	9
VI	Rush 02	10
VII	Rush 03	11
VIII	Rush 04	12
IX	제출 및 동료 평가	13

Chapter I

Instructions

- 해당 과제는 자동으로 동료평가에 등록됩니다.
- 취소하지 마세요. 두 번째 기회는 없습니다..
- 과제에 관한 모든 질문들은 과제를 더욱 복잡하게 만들 것입니다.
- 모든 연습문제의 제출은 제출 절차에 따라 진행되어야 합니다.
- 파일 제출 직전까지도 이 문서가 변경될 수도 있으니 주의하세요!
- 프로그램은 다음 명령어와 플래그로 컴파일해야 합니다. : cc -Wall -Wextra -Werror;
- 만약 당신의 프로그램이 컴파일되지 않는다면 0점을 부여받습니다.
- 프로그램은 Norm에 따라 작성되어야 합니다. 만약 보너스 파일/함수가 있는 경우 그것들도 Norm 검사 대상에 포함되며 Norm Error가 있을 경우 과제 점수는 0점입니다.
- 오류 처리는 일관성이 있어야 합니다. 자유롭게 오류 메시지를 출력하거나 깔끔하게 사용자에게 제어권을 반환하세요.
- Rush의 연습 문제들은 2~4인으로 수행해야 합니다.
- 당신의 팀이 진행해야 하는 필수 연습 문제(mandatory)의 rush 번호는 다음과 같은 규칙으로 선정됩니다. :
팀장의 로그인(인터넷 닉네임)의 첫 글자 알파벳 인덱스(1~26까지)의 나머지 연산 5 (자세한 내용은 groups_subject.txt 참고)
- 당신은 자동으로 구성된 팀원들과 함께 프로젝트를 수행하고 모든 팀 구성원과 함께 디펜스에 참가해야 합니다.
- 디펜스를 할 때에는 이미 당신의 프로젝트가 완료된 상태여야 합니다. 디펜스의 목적은 당신이 당신의 작업물을 발표하고 설명하는 것입니다.
- 팀의 각 구성원은 프로젝트의 작업물에 대해 완전히 알고 있어야 합니다. 만약 분업으로 진행했다고 하더라도 본인의 작업 영역만이 아닌 모든 영역을 이해해야 합니다. 디펜스를 진행하며 모든 팀 구성원은 질문 받을 것이며 과제의 최종 점수는 가장 어설픈 설명을 기준으로 부여됩니다.

- 말할 것도 없지만, 팀원을 모으는 것은 당신의 의무이며 책임입니다. 전화, 이메일, 전서구, 텔레파시 등 팀원들과 연락할 수 있는 모든 수단을 동원하세요. 그러니 수고스럽게 해명을 할 필요는 없다는 의미입니다.
(단, 서울 캠퍼스 한정으로 예외사항이 존재 할 경우 사전 공지로 안내합니다.) 인생은 공평하지 않습니다(Life is not fair). 원래 그런거에요.
- 하지만, 정말로 꾸준하게 모든 수단과 방법을 가리지 않았음에도, 그리고 그것을 증명 할 수 있다면 연락이 두절된 팀원이 존재하더라도 일단 과제를 수행하세요. 디펜스를 진행하는 동안 우리가 확인 해 보겠습니다. 연락 두절 대상이 팀장인 경우에도 당신들은 제출 레포지터리에 접근 가능하며 디펜스를 진행할 때 대응이 가능합니다.
- 다른 연습문제(들)를 추가로 제출하거나 필수 연습 문제(mandatory)에 프로그램 인수(program arguments)를 사용하는 기능을 추가하여 디펜스에 성공한다면 각각 보너스 점수를 획득할 수 있습니다.
다시 한번 강조합니다만, 보너스에서 Norm Error가 발생할 경우 필수 연습 문제(mandatory)가 아무리 완벽해도 0점이며 이는 타협 불가합니다.



보너스를 고려하기 전에 팀에 원래 할당된 필수 연습 문제가 완벽하게 완성되었는지 확인하세요. 보너스가 아무리 대단하고 완벽하더라도 필수 연습 문제가 0점이면 그냥 0점입니다.

Chapter II

Foreword

다음은 누구나 알 법한 유명 TV 방영물의 주제가 가사입니다. :

[1절]

자~ 이제 시작이야 (내 꿈을)
내 꿈을 위한 여행 (피카츄)
걱정따윈 없어 (없어)
내 친구랑 함께니까

처음 시작은 어색할지도 몰라 (몰라)
내 친구 피카츄
날 지켜줄 거라고 믿고 있어
누구라도 얍보다간 큰일나

언제나 어디서나 (피카츄가 옆에 있어)
약할 때나 강할 때나 (피카츄가 옆에 있어)
너와 난 함께라면 우린 최고야

언제 언제까지나 진실한 마음으로
언제 언제까지나 하나가 되어
언제 언제까지나 최고가 되는
언제 언제까지나 그날을 위해

(피카츄 피카츄 츄 츄 피카 피카 피카츄 피카츄)

나는 지금 당신이 노래를 부르고 있다고 장담할 수 있지만, 뭐 중요한 것은 아닙니다.
중요한 것은 이 과제와 포켓 몬스터는 전혀 관련이 없다는 것입니다.

Chapter III

Main subject

	Exercise 00
	Rush0X
Turn-in directory :	<i>ex00/</i>
Files to turn in :	<code>main.c, ft_putchar.c, rush0X.c</code>
Allowed functions :	<code>write</code>

- 제출 파일은 `main.c, ft_putchar.c` 그리고 `rush0X.c`이며, '0X'는 Rush 문제의 번호를 나타냅니다. 예시: `rush00.c`
- 이 세개의 파일이 함께 컴파일됩니다.
- `ft_putchar.c` 파일에는 `ft_putchar` 함수가 포함되어야 합니다.
- `main.c` 의 예시를 봅시다. :

```
int main()
{
    rush(5, 5);
    return (0);
}
```

- 각각 x 및 y라는 이름의 int 타입의 두 변수를 인수로 사용하는 `rush` 함수를 작성해야 함을 의미합니다. 이 함수가 `rush0X.c` 파일에 있어야 하는 것을 설명해야 할까요?
- 함수 `rush`는 가로변 문자 x개, 세로변 문자 y개로 구성된 직사각형을 화면에 출력합니다.
- 당신의 함수는 절대로 충돌(crash)이 발생하거나 무한 반복이 발생하면 안 됩니다.
- 디펜스 중에 당신의 `main`을 수정하여 프로그램의 완성도를 점검 할 것입니다. 예를 들면 이렇습니다:

```
int      main()
{
    rush(123, 42);
    return (0);
}
```

Chapter IV

Rush 00

- rush(5,3) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
o---o
|
o---o
$>
```

- rush(5, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
o---o
$>
```

- rush(1, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
o
$>
```

- rush(1, 5) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
o
|
|
|
|
$>
```

- rush(4, 4) 의 실행 결과는 다음과 같습니다.:

```
$> ./a.out
o--o
| |
| |
o--o
$>
```

Chapter V

Rush 01

- rush(5,3) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
/**\
 *
\*/
$>
```

- rush(5, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
/**\
$>
```

- rush(1, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
/
$>
```

- rush(1, 5) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
/
*
*
*
*
\
$>
```

- rush(4, 4) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
/**\
 *
 *
\*/
$>
```

Chapter VI

Rush 02

- rush(5,3) 의 실행 결과는 다음과 같습니다. :

```
$>./a.out
ABBA
B B
CBBC
$>
```

- rush(5, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBA
$>
```

- rush(1, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
$>
```

- rush(1, 5) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
B
B
B
C
$>
```

- rush(4, 4) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBA
B B
B B
CBBC
$>
```

Chapter VII

Rush 03

- rush(5,3) 의 실행 결과는 다음과 같습니다. :

```
$>./a.out
ABBC
B B
ABBC
$>
```

- rush(5, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBC
$>
```

- rush(1, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
$>
```

- rush(1, 5) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
B
B
B
A
$>
```

- rush(4, 4) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBC
B B
B B
ABBC
$>
```

Chapter VIII

Rush 04

- rush(5,3) 의 실행 결과는 다음과 같습니다. :

```
$>./a.out
ABBC
B B
CBBA
$>
```

- rush(5, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBC
$>
```

- rush(1, 1) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
$>
```

- rush(1, 5) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
A
B
B
B
C
$>
```

- rush(4, 4) 의 실행 결과는 다음과 같습니다.:

```
$>./a.out
ABBC
B B
B B
CBBA
$>
```

Chapter IX

제출 및 동료 평가

평소처럼 git 저장소에 과제를 제출하세요. 디펜스 중에는 저장소 내부의 작업만 평가됩니다. 파일 이름이 올바른지 다시 한 번 확인하는 작업을 주저하지 마세요.



이 프로젝트의 문서에서 요구한 파일만 반환해야 합니다.