2주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

**q1**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**q2**

텍스트, 메뉴, 스크린샷, 문서이(가) 표시된 사진

자동 생성된 설명텍스트, 메뉴, 스크린샷, 문서이(가) 표시된 사진

자동 생성된 설명

**q3**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**2.**

**1. main**

1. 프로그램을 시작할 때 매개변수를 검사하면서 파일이름을 받는다.
2. 만약 매개변수 개수가 2가 아니면 사용 방법에 대한 안내 메시지를 출력한다.
3. 파일을 읽기모드로 열고, 파일이 열리지 않으면 오류 메시지를 출력한다.
4. 동적 할당하여 문자열을 저장하기 위한 공간을 만든다.
5. 파일로부터 첫 line을 받는다. 만약 파일이 비어 있다면 중단한다.
6. Remove\_Blanks\_At\_The\_End 함수를 호출해서 첫 line뒤에 있는 공백 문자들을 제거한다.
7. Print\_Line함수를 호출해 첫 line을 출력 형식에 맞게 출력한다.
8. 파일의 두 번째 line부터 순서대로 읽어 들여 처리한다.
9. Remove\_Blanks\_At\_The\_End 함수로 각 줄의 뒤에 있는 공백 문자들을 제거하고, Print\_Line함수로 특정 조건에 따라 출력한다.
10. 파일의 마지막 line이 빈 줄이 아닌 경우, 마지막 문자가 줄 바꿈 문자인지 확인하고 필요에 따라 출력한다.

**2. Remove\_Blanks\_At\_The\_End**

1. 문자열의 끝부터 시작하여 줄 바꿈 문자가 나올 때까지 탐색한다.
2. 줄 바꿈 문자이전에 공백이 아닌 문자를 찾아서 그 사이에 있는 공백을 제거한다.
3. 만약 마지막 문자가 줄 바꿈이라면 줄 바꿈 문자와 공백을 추가하지만 아니라면 NULL 문자를 추가하여 문자열을 종료한다.

**3. Print\_Line**

1. Get\_Blanks\_Chars 함수를 호출해 문자열의 공백과 문자의 개수를 계산한다.
2. 계산된 공백과 문자의 개수를 기반으로 문자열을 출력하는데 출력할 때에는 각 제한에 따라 형식에 맞게 출력한다.
3. 출력이 끝난 후에는 다음 출력을 위해 시작 위치와 카운터를 업데이트한다.

**4. Get\_Blanks\_chars**

1. 문자열을 탐색하면서 공백과 문자의 개수를 계산한다.
2. 공백의 개수를 셀 때는 이미 공백이 탐지되었는지 확인하면서 중복 계산을 방지한다.
3. 문자의 개수를 계산하면서 줄 바꿈이나 NULL 문자를 만나면 계산을 종료한다.

**3.**

q1의 Makefile

CC = gcc // c 컴파일러 매크로 정의

Target = animal // 실행파일 이름 매크로 정의

Objects = dog.o blackcow.o turtle.o main.o // 목적파일 이름 매크로 정의

$(Target) : $(Objects) // 매크로 변환

$(CC) -o $@ $^ // $@ : 현재 Target 이름 , $^ : 현재 Target이 의존하는 대상 전체 목록

.PHONY : clean // clean 파일이름과의 충돌 방지

clean :

rm $(Targets) $(Objects) // make clean 명령어로 지워지는 파일들

q3의 Makefile

cc=gcc // c 컴파일러 매크로 정의

cflags = -W -g // -W : 경고를 보내는 옵션 , -g : 디버깅 정보를 포함

target = main // 실행파일 이름 매크로 정의

objects = main.o String\_Manipulation.o Output.o // 목적파일 이름 매크로 정의

$(target) : $(objects) // 매크로 변환

$(cc) $(cflags) -o $(target) $(objects) // 컴파일 명령어

%.o : %.c // c파일을 목적파일로 변환

$(cc) $(cflags) -c -o $@ $< // $< : 현재 Target이 의존하는 대상 중 변경된 것들 목록

main.o string\_manipulation.o output.o : Header.h // 목적파일들은 Header.h에 종속

.PHONY : clean // clean 파일이름과의 충돌 방지

clean :

rm $(target) $(objects) // make clean 명령어로 지워지는 파일들

**4.**

Print\_Line 함수에서 B\_Flag를 통해 문자열의 첫 문자가 공백인지 확인한다.

만약 B\_Flag가 1이고(문자열의 첫 문자가 공백), 현재 문자열이 LIMIT을 초과하지 않으면

새로운 line이 아니라 현재 line에 출력한다.

하지만 B\_Flag가 0이거나 현재 문자열이 LIMIT을 초과한다면 새로운 line으로 넘어간다.

이때 첫 부분의 공백은 공백 개수만큼 출력한다.

Get\_Blanks\_Chars 함수를 통해 문자열에 대한 공백과 문자 수를 확인한다.

**5.**

-C dir : Makefile을 읽기 전에 디렉토리를 변경

-d : 디버깅 정보를 출력

-e : environment 변수를 Makefile 변수보다 우선시함

-f file : file을 Makefile로 읽음

-h : Makefile이 이해하는 Option 출력하고 종료

-i : 모든 error 무시

-I dir : Makefile을 찾기 위해 디렉토리 변경

-j : 동시에 실행하는 작업 개수 지정

-k : error 발생해도 가능한한 지속 실행

-l : 평균 부하가 load(부동소숫점)라면 새로운 작업을 시작하지 않음  
-n : 실행될 명령은 출력하지만 실행하지는 않음

-o file : 어떤 file을 재 컴파일하는 것을 방지

-p : Makefile을 읽은 결과의 Database를 출력

-q : 어떤 명령도 실행하지 않고, 어떤 것도 출력하지 않음. 오로지 종료 상태 값만 반환

-r : 내장 묵시적 규칙들의 사용을 제거  
-s : 명령들이 출력되지 않음

-t :file들의 날짜 시간만 현재 시간으로 마크하여 make 추후 실행을 방지

-v : make 프로그램의 Version, 저작권 등을 출력

-w : Makefile을 실행하기 전후에 작업 디렉토리 메시지를 출력

-W file : Target file이 변경된 것처럼 가장