

과제 1번

< 사용 변수 >

input는 char[10000]형 배열로 사용자의 입력이 저장됩니다. input2는 input 원본 유지용 배열이고, norepeatword는 chara [200][50] 배열로, 중복되지 않은 단어를 저장하기 위한 배열입니다. word는 단어 수, line은 줄 수, used는 원본 길이를 저장하기 위해, norepeat은 중복되지 않은 단어 수를 저장하는 배열로, 모두 int 형입니다. diffwordcounter 함수의 nosameword는 중복되지 않은 단어 수를 저장하는 변수이고, token은 char* 변수로 strtok()에 사용됩니다. finder는 중복 단어를 세는 함수입니다.

< 풀이 방법 >

[main 함수]

- 먼저 사용자가 end를 단독으로 입력할 때까지 fgets를 통해 입력 받습니다. used 변수와 strncpy를 활용해 입력된 줄(enter 기준)을 input2 변수에 복사합니다. \n 여부를 strchr()을 통해 확인 해 line 변수에 줄 수를 기록하고, strtok를 통해 입력받은 줄의 단어 수를 셉니다. end 입력으로 입력 종료된 이후, diffwordcounter 함수로 중복단어를 제거하고, 입력된 문장/단어/중복단어제거단어 수를 출력하고, qsort로 단어를 정렬해 출력합니다.

[diffwordcounter 함수]

- strtok()을 사용해 단어를 나누고, 중복 단어가 없는 경우 norepeatword[]에 저장합니다. 토큰이 norepeatword[]와의 비교(strcmp)에서 동일(0)이 나오는 경우, finder++을 하고 norepeatword[]에는 저장하지 않습니다. 마지막으로 중복제거 단어수인 nosameword를 반환합니다.

[compare 함수 + qsort()]

- qsort에 사용하는 compare함수 입니다. strcmp((char*)a,(char*)b)를 해 내림차순 정렬을 했습니다.

원하는 문장을 입력하세요.
end를 입력하면 입력이 종료됩니다.
once when i was six years old i
saw a magnificent picture
end

입력된 문장의 수: 2
입력된 단어의 수: 12
중복이 제거된 단어의 수: 11

a
i
magnificent
old
once
picture
saw
six
was
when
years

20224382 안선우

원하는 문장을 입력하세요.
end를 입력하면 입력이 종료됩니다.
beta gamma delta
alpha omega beta zeta epsilon
sigma lambda zetaa lambda
end

입력된 문장의 수: 3
입력된 단어의 수: 12
중복이 제거된 단어의 수: 10

alpha
beta
delta
epsilon
gamma
lambda
omega
sigma
zeta
zetaa

20224382 안선우

과제 2번

< 사용 변수 >

char[5][52]형의 answer 변수는 랜덤으로 생성되는 문제를 저장하는 변수이고, 유저의 입력은 char user[5][52]에 저장됩니다. char 형의 blankroutine은 몇개의 단어마다 띄어쓰기가 나오는지 저장하는 변수이고, wordcount는 생성된 단어 수 중 틀린 단어수를 제외한 회차 당 결과를 저장하는 변수이며 score는 int형으로 최종 점수를 저장하는 변수입니다. start, end는 걸린 시간을 체크하기 위한 clock_t형 변수이고, double 형의 spentsec에 걸린 시간을 저장합니다.

< 풀이 방법 >

[main 함수]

- 먼저 rand()를 사용해 4-10개 단어 간격으로 빈칸을 만들고, 40-50개 사이의 랜덤한 대문자 글자를 만듭니다. 이후 for문과 blankroutine을 활용해 띄어쓰기를 포함해 출력하고, fgets()를 사용해 사용자의 입력을 받습니다. 이후 strcmp()과 strchr()을 사용해 틀린 부분에는 V를 표시하고, 옳게 입력한 문자수 / 걸린 시간 * 60 을 통해 분당 타자수를 출력하고 score 변수에 추가합니다. 마지막에 총 점수를 출력했습니다.

(1/5) 다음을 입력하세요

```
VXDILNXRM XPBFRIDS ZGSVNJXF PIGBCGSN FN
VXDILNXRM XPBFRIDS ZGSCNJXF PIGVXGSN FN
                        V      VV
```

129자/분

(2/5) 다음을 입력하세요

```
KKDPWLCVAY ZEUXXUREK BJVAKEZFY ZROTMOFPA JVDWQUP
KKDPWLCVAY ZEUXXUREK BJVAKEZFY ZROTMOFPA JVDWQUP
```

157자/분

(3/5) 다음을 입력하세요

```
GAWSLEKTF KRUXABIX PLNVXFIV QWWGHMUC AA
ASDFGHKLK KRUZABIX PLNZXFIV QWW HMUC AA
VVVVVV VV      V      V      V
```

82자/분

(4/5) 다음을 입력하세요

```
TPGVLLJT AQSVMXS SNSXZMJ UGAZKFK WRBOJVG MK
TPFVLLJT AQSVMXS SNSXZMJ UGAZKFK WRBOJVG MK
      V
```

135자/분

(5/5) 다음을 입력하세요

```
XJHTKGYG HDUBHDA DXKCXMC VJRRHCC BGMCMCQ
ZJHTKGYG HDUCHAA DZKKXMC VJRRHCC BGMCMCQ
V      V V      V V
```

98자/분

당신의 점수는 601점 입니다.

20224382 안선우

과제 3번

< 사용 변수 >

- x, y, add, sub은 모두 char 형 배열로, x, y는 char [82], add, minus는 char [83] 형입니다. char minus는 부호 여부를 파악하는 변수로, 1이면 부호를 출력, 0이면 출력하지 않습니다.
- adder 함수에서 char 형 lenx는 전달받은 x의 길이를, leny는 y의 길이입니다. length는 (더하기가 길이가 더 긴 숫자문자열을 기준으로 진행되기에) x, y 중 더 긴 문자열의 길이를 저장하는 변수이고, temp는 더하기 결과를 잠시 저장하는 char형 변수입니다.
- subtractor 함수에서도 처음에는 char형 lenx와 leny는 x와 y의 길이를 나타내나, 이후 lenx는 더 큰 수를 가진 배열의 길이를, leny는 더 작은 수를 가진 배열의 길이를 나타내게 됩니다. char형 minus는 반환되는 값으로, 부호 표현 여부를 나타냅니다(1=표현/0=없음). checkflag는 10을 빌려왔는지 체크하는 char형 변수이고, left와 right는 char*형으로, left가 더 큰 수를 가진 배열(x or y)의 포인터를, right가 더 작은 수를 가진 배열의 포인터를 가르칩니다.
- reverse 함수에서 len은 int형으로, 전달받은 문자열의 길이를 저장합니다. temp는 char 형으로 문자열을 뒤집는 과정에서 잠시 값을 저장하는 역할을 합니다.

< 풀이 방법 >

[main 함수]

- fgets()를 사용해 사용자에게 x와 y를 입력받습니다. 이후 입력된 값의 %0 전에 저장된 %n을 %0으로 수정하는 과정을 거치고, adder함수와 subtractor 함수를 실행해 덧셈 뺄셈 값을 출력합니다.

[adder 함수]

- lenx와 leny 중 더 큰 값을 length 변수에 저장하고, 이를 기준으로 덧셈을 실행합니다. x와 y를 reverse 함수를 통해 일의자리~마지막자리 순으로 뒤집은 후, for문을 사용해 length 만큼 한자리수 덧셈을 실행합니다. length가 긴 문자열의 길이인 만큼 짧은 문자열의 범위를 넘어서 이후엔 임의로 0으로 간주해 더하기를 진행합니다. 더한 내용을 10으로 나눈 나머지만큼을 add 배열에 저장하고, 몫은 다음 덧셈에 적용될 수 있도록 합니다. add의 마지막에 %0가 들어갈 수 있도록 하고, reverse를 통해 add 배열 또한 가장 큰 자릿수부터 표현될 수 있도록 했습니다.

[subtractor 함수]

- adder함수에서 reverse한 내용을 다시 reverse했고, strcmp를 통해 left가 더 큰 수를 가진 문자열을 가리키도록 했습니다. 이후, 다시 reverse 해 작은 자릿수의 숫자가 앞에 오도록 둔 후, 뺄셈을 진행합니다. 이번에도 짧은 문자열의 범위를 넘어서 이후에는 임의로 0으로 간주하고 뺄셈을 진행하며, 앞 자릿수에서 10을 빌려오는 경우 checkflag를 통해 다음(앞 자릿수 뺄셈) 단계에 적용되도록 했습니다. 이번에도 sub의 마지막에 %0가 들어갈 수 있도록 했고, reverse를 통해 sub 배열이 가장 큰 자릿수부터 표현되도록 했습니다.

[reverse 함수]

- 문자열을 뒤집는 함수입니다. ori[i]의 값을 ori[len-i-1]의 값과 바꾸기에 마지막에 있는 %0은 영향을 미치지 않습니다.

123456789123456789123456789123456789123456789123456789123456789 +
98765432198765432198765432198765432198765432198765432198765432

[illegible]

123456789123456789123456789123456789123456789123456789123456789 -
98765432198765432198765432198765432198765432198765432198765432

-86419753286419753286419753286419753286419753286419753286419753286419753286419753286419753

[illegible]

과제 4번

< 사용 변수 >

- char [1000]형의 input 변수는 사용자가 입력하는 식을 전달받는 변수이고, input2는 input 변수를 복사하는, 원본저장용 변수입니다. strtok()를 사용하기에 char *형 temp 변수가 존재하고, int형 result 함수는 최종 결과값을 저장하는 변수입니다.

< 풀이 방법 >

[main 함수]

- strtok()를 사용해 +, -, = 기호를 제거하고 토큰화 해줍니다. 이후, 해당 토큰 앞에 있던 기호가 -인 경우, result에서 빼주어 빼기를 적용했습니다. atoi는 temp가 가리키는 문자열을 숫자로 변경해주는 함수이기에 덧셈/뺄셈이 가능하게 됩니다. 마지막으로 결과를 출력했습니다.

