프로그래밍 과제 3

20222663 권기영

1. 단어 정렬 - word_sorting.c

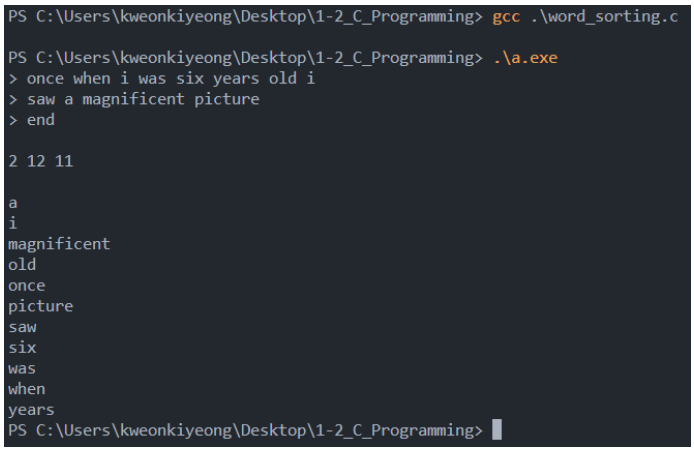
<구현>

1) 입력은 fgets로 한줄씩 받고, end를 입력할 때까지 계속해서 입력을 받는다.

2) 입력한 한 줄에 대해서 strtok 함수를 사용해 공백문자를 기준으로 토큰화를 하고, 단어를 저장하는 배열에 토큰이 있는지 없는지를 직접 만든 find 함수를 통해 판단한 뒤, 배열에 토큰이 없다면 저장한다.

2) 정렬은 qsort로 하고, compare 함수는 strcmp에 (char *)형으로 타입캐스팅한 인자를 넣어 비교를 진행했고, 사전순으로 정렬하였다.

<실행 예>



end를 입력받을 때 까지 문장을 입력받고, end를 입력받으면 순서대로 문장 개수, 입력 단어 개수, 저장 단어 개수를 출력한다.

또한 사전순으로 저장한 단어를 정렬한 뒤 출력한다.

2. 타자 게임 - typing_game.c

<구현>

1) gererateStr함수: 무작위의 문장을 만드는 함수이다. 문장과 단어 배열을 인자로 받아서 글자 개

수(4 ~ 10)를 랜덤으로 정하고, 글자 개수에 맞는 단어에 들어가는 알파벳(A ~ Z)을 랜덤으로 정한다. 무작위로 만들어진 단어들을 strcat 함수를 활용해서 문장의 뒤에 계속 연결한다. 만약 현재 문장의 길이 + 새로 추가될 글자 개수 가 50을 넘는다면 break 한다.

2) evalTyping함수: 사용자가 문장에 맞게 타자를 칠 때, 타자 속도를 반환하고 틀린 부분을 출력해주는 함수이다. 문자를 하나하나 비교하면서 같으면 점수를 1점 올리고, 다르면 그 자리에 오류를 출력한다. 오류를 출력할 때 주의해야 할 점은, 입력한 문장이 길이가 더 짧거나 더 길면, 그 부분은 무조건 오류로 표시해야 한다. 리턴값은 분당 타자 속도를 리턴한다.

3) 총 5번의 시행을 거친 뒤, 최종 점수를 출력한다.

<실행 예>

```
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> gcc .\typing_game.c
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> .\a.exe
게임을 시작합니다.
(1/5) 다음을 입력하세요
= ICSGQMSUG POJQICHW TCUTRB TMNTQHTEBF XXTCSK
= ICSGQMSUG POJQICHW TCUTRB TMNTQHTEBF XXTCSK

592타/분
(2/5) 다음을 입력하세요
= UJLRTTBU NQRVP JFNTN HOHQL BDEDRLK RRBEGJ
= UJLRTTBU NQRVP JFNTN HOHQL BDEDRLK abcdGJ
                                    ^^^^
307타/분
(3/5) 다음을 입력하세요
= TDCOO BFIZXLOAT JJIWIYUZFH YWXH IIZCYFF PTVNWUA
= TDCOO BFIZXLOAT JJIWIYUZFH YWXH IIZCYFF PTVNWUAasdf
                                                ^^^^
192타/분
(4/5) 다음을 입력하세요
= GQZWQFJL QPPHZD TKCCUQCQKA QTZHR GCPQDKDZ
=
  ^^^^^^^^ ^^^^^^ ^^^^^^^^^^ ^^^^^ ^^^^^^^^
43타/분
(5/5) 다음을 입력하세요
= JYSIWOD MXOWQT VBSNWYYRQ CXNRMAJX YAZOB MOKQVA
= JYSIWOD MXOWQT VBSNWYYRQ CXNRMAJX YAZOB MOKQVA
                                              ^^^^^
263타/분
당신의 점수는 1397점 입니다.
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming>
```

1번 시행은 모두 동일하게 쳐서 오류를 출력하지 않는다.

2번 시행은 다른 것들에 대해서 오류를 출력하고 있다.

3번 시행은 주어진 문장을 초과한 입력을 했기 때문에 오류를 출력한다.

4번 문장은 전부 공백을 입력하여 공백을 제외한 모든 곳에서 오류를 출력한다.

5번은 문장과 동일하게 쳤지만, 마지막에 공백 5번을 추가로 입력하여 오류를 출력하는 모습이다.

3. 80자리 계산기 - bignum_cal.c

<구현 아이디어>

1) 더하기와 빼기 모두 사람이 계산하는 알고리즘을 통해 계산하도록 한다. 즉, 숫자를 하나씩 연산하면서 그 다음 자리수에 대한 빌림이나 올림을 수행하는 방법이다.

2) 숫자를 저장할 때 역순으로 저장한다. 자리수 올림이나 쓸모없는 0 지우기의 작업이 숫자의 가장 높은 자리수에서 일어나는데, 역순으로 저장하지 않으면 배열의 시작 위치에서 삽입, 삭제 과정을 진행하고, 이는 배열 전체를 밀어야 하므로 비효율적이다. 따라서 역순으로 저장 후 배열의 끝 위치에 삽입한다.

3) Add 함수: 같은 자리의 숫자끼리 더하고, 그 값이 10 이상이면 한 칸 윗자리 수에 1을 올림해준다. 숫자의 길이가 차이 날 때는 더 짧은 쪽의 값을 0으로 취급한다.

4) Subtract 함수: 같은 자리의 숫자끼리 빼고, 그 값이 0보다 작으면, 한 칸 윗자리 수에서 1을 빌려온다. 두 개의 피연산자 중 더 큰 값을 판단해야 하는데, 그 이유는 작은값 - 큰 값의 연산을 - (큰값 - 작은값)의 연산으로 바꿔서 좀 더 쉽게 뺄셈 연산을 수행하기 위함이다.

5) 뺄셈 연산이 끝나면 높은 자리수에 있는 쓸모없는 0을 제거한다. 예를 들어 결과값이 0001이라면, 쓸모없는 0을 지우고 1을 출력하기 위함이다.

<실행 예>

```
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> gcc .\bignum_calculation.c
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> .\a.exe
input X. 124513453245
input Y. 121524523434

X + Y = 246037976679
X - Y = 2988929811

input X. 131412412
input Y. 96374573457

X + Y = 96505985869
X - Y = -96243161045

input X. 1
input Y. 999999999999999

X + Y = 1000000000000000
X - Y = -999999999999998

input X.
input Y.
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming>
```

```
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> python
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 124513453245 + 121524523434
246037976679
>>> 124513453245 - 121524523434
2988929811
>>> 131412412 + 96374573457
96505985869
>>> 131412412 - 96374573457
-96243161045
>>> 1 + 999999999999999
1000000000000000
>>> 1 - 999999999999999
-999999999999998
>>>
```

python에서 계산한 결과와 동일함을 확인할 수 있었다.

4. 계산식 - many_operand_cal.c

<구현 아이디어>

1) 입력을 fgets로 전부 받은 뒤, 연산자의 정보를 저장하는 별도의 배열을 하나 만든다. 연산자가 나온 순서대로 배열에 저장하고, +면 1을, -면 -1을 저장한다.

2) 그 후 +와 -를 기준으로 토큰화를 진행한다. 토큰화를 하면서 연산자 정보 배열의 index도 같이 움직이면서 연산의 종류를 결정해 연산한다.

3) 피연산자의 값이 10000 이하의 양의 정수로 제한되었기 때문에, 문자열 연산보다는 atoi를 활용해 정수 연산을 진행하는 것이 훨씬 합리적이다.

<실행 예>

```
PS C:\Users\kweonkiyeong\Desktop\1-2_C_Programming> .\a.exe
input operation > 12+34-56+789-123-56+78
= 678
```

연산자 배열에 (1, -1, 1, -1, -1, 1)을 순서대로 저장하고, 토큰화를 하면서 연산자에 맞는 연산을 수행하고, 올바른 출력 값이 나온다.