

計算機入門及び演習

C 言語レポート課題__文字列、文字コード、標準入出力

6 3 2 1 1 2 0

横溝 尚也

提出日：1 2 月 2 1 日 (火)

1 レポート課題1

入力された文字列にアルファベット、数字、その他の文字が、それぞれ何文字含まれているのかを出力する。

以下の点については考慮しなくて良い。

- 空白を含む文字列が入力される。
- 文字列が入力されない。

2 アルゴリズムの説明

アルファベット、数字、その他の文字がそれぞれ何文字含まれているのか数える。アルファベット、数字、その他の文字の3種類に入力された文字を識別してカウントし、その文字数を出力すればよい。その際、入力された文字数は配列に格納し、格納された文字に対して識別を行えばよい。

3 プログラムの説明

```
1  #include <stdio.h>

2int main(){
3    char sequence[10000];
4    scanf("%s" , sequence);

5    int numcount = 0;
6    int abccount = 0;
7    int markcount = 0;
8    int i;

9    for (i = 0; sequence[i] != '\0' ; i++){
10        if ( (sequence[i] >= 'a' && sequence[i] <= 'z' ) || (sequence[i] >='A' && sequence[i] \
11            ] <= 'Z') ) {
12            abccount++;

13        }else if (sequence[i] >= '0' && sequence[i] <= '9'){
14            numcount++;

15        }else{
16            markcount++;

17        }
```

```

18     }

19     printf("アルファベット:%d 文字\n" , abccount);
20     printf("数値:%d 文字\n" , numcount);
21     printf("その他:%d 文字\n" , markcount);

22}

```

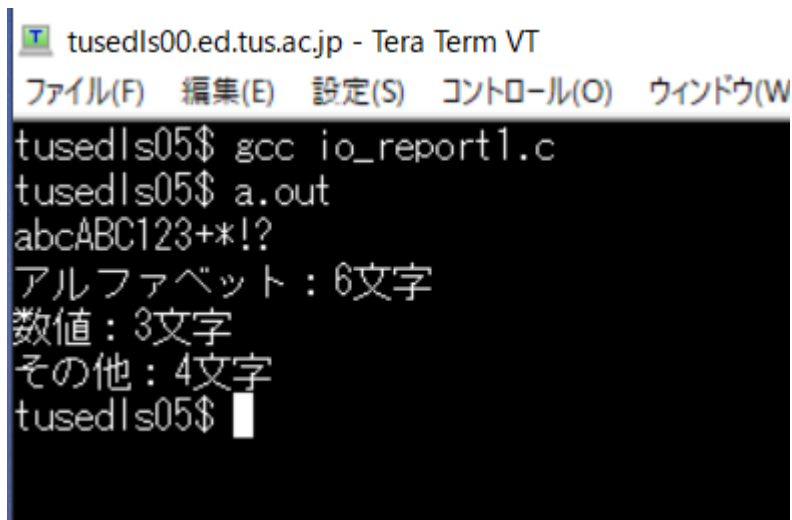
3 行目では外部から入力された文字列を格納するための配列を宣言、4 行目では外部から入力された文字列を格納するための関数 `scanf("")` を宣言している。

5 から 7 行目ではアルファベット、数字、その他の文字列の 3 種類の文字をそれぞれ数え上げるための整数型変数の宣言と代入をしている。8 行目では 9 行目以降で用いるくり返し文のくり返し範囲を定めるための変数を宣言している。

9 から 18 行目では繰返し範囲を外部から入力された文字列すべての文字の種類を判別しカウントするまでとし、それぞれの文字の種類の際にその文字の種類をカウントするという処理を行っている。

19 から 21 行目で数え上げた文字数を出力している。

4 実行結果



```

tusedls00.ed.tus.ac.jp - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W)
tusedls05$ gcc io_report1.c
tusedls05$ a.out
abcABC123+*!?
アルファベット : 6文字
数値 : 3文字
その他 : 4文字
tusedls05$

```

図 1 実行結果 (1)

5 考察

このプログラムの優れている点は、アルファベットやその他の記号だけでなく数字も `char` 型の変数として宣言し、配列に格納しても `if` 文を用いて記号の種類を識別して分類している点である。`if` 文の条件式で

$sequence[i] \geq 'a' \quad || \quad sequence[i] \leq 'z'$ とすることで a から z のアルファベットに当てはまる条件式となっている。文字型変数とすればこのように区別することができ、簡潔にアルファベットを数えることを可能としている。

6 レポート課題2

入力された演算子と数値、そしてそれまでの計算結果を使用した計算を実行し、その結果を出力する。

例えば、初期状態で「+ 10」と入力されると「0+10」を計算し「10」と出力する。

更に「- 20」と入力すると「10-20」を計算し「-10」を出力する。

詳細は以下の通り。

(1) 入力が終了するまでプログラムは動作し続ける。

(2) 初期値は 0 とする。

(3) 入力可能な演算子は以下の通り

+ 加算

- 減算

乗算

/ 除算

(4) 演算子と数値は以下のように入力する。

+(空白もしくは改行)10

(5) 以下の場合には計算をせずにエラーメッセージを出力する。

- ゼロ除算になる。
- 演算子を入力する箇所に演算子以外の文字が入力されている。
以下の点については考慮しなくて良い。
- 数値が、プログラム上で表現可能な範囲を超える。
- 計算機やプログラミング言語の問題で計算結果が正しくならない。
- その他、上述していない入力の誤りがある。

7 アルゴリズムの説明

”入力された演算子と数値から計算結果を出力する”つまり、`scanf("")` 関数を用いて数値と演算子を入力し、それをもとに計算を行えばよい。

演算子が適切か、また 0 助産でないのか判別するには `if` 文を用いてエラーの出力を行えばよい。

”また計算を行う際に、入力が終了するまで計算を行う”とあるので `while` 文を用いて

`while(scanf("%c%*c%f%*c",&operation,&num) != EOF)` と繰り返し範囲を設定すればよい。

8 プログラムの説明

```
1  #include <stdio.h>

2 int main(){
3     float num;          /*整数型変数 num の宣言*/
4     char operation;     /*文字型変数 operation の宣言*/
```

```

5    float answer = 0; /*整数型変数 answer の宣言と代入*/

6    while( scanf("%c%c%f%c" , &operation , &num ) != EOF ){
/*外部から入力された数値と演算子を読み込み、処理を繰り返す*/
7        if(operation == '/' && num == 0){
/*ゼロ除算となるときのエラーの出力*/
8            printf("エラー：ゼロ除算になる。\\n");

9        }else if (operation != '+' && operation != '-'
/*演算子が適切でないときのエラーの出力*/
                && operation != '*' && operation != '/'){
10            printf("エラー：演算子ではない。\\n");

11        }else{
/*正常に外部から入力されたときの処理*/
12            switch(operation){
/*それぞれの演算子を読み込まれたときの answer に行う処理*/
13                case '+':
14                    answer = answer + num;
15                    break;

16                case '-':
17                    answer = answer - num;
18                    break;

19                case '*':
20                    answer = answer * num;
21                    break;

22                case '/':
23                    answer = answer / num;
24                    break;
25            }

26            printf("結果：%f\\n" , answer );
/*計算結果の出力*/
27        }

28    }

```

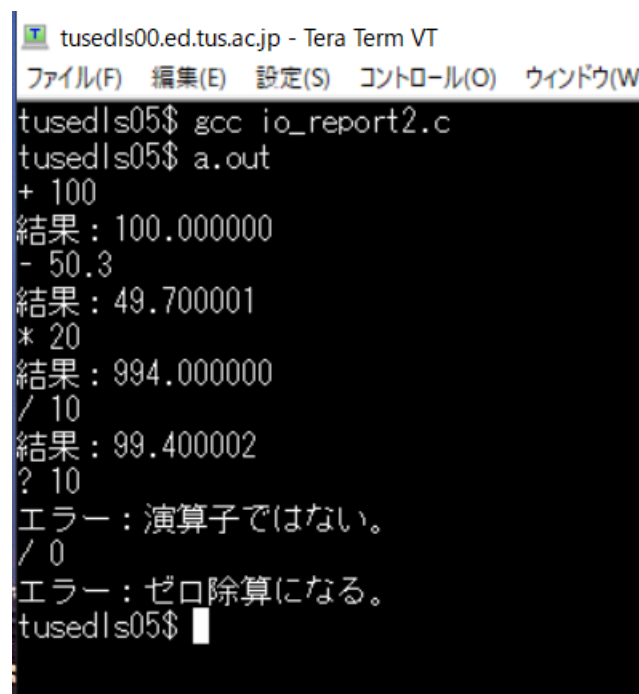
29}

3,4 行目で少数型変数 `num` と文字型変数 `operation` を入力する数値と演算子用の変数を宣言、5 行目では計算結果を格納するための少数型変数 `answer` を宣言している。

6 行目以降で計算処理を行うくり返し文である。条件式は演算子と数値の入力が終わるまでとしてプログラムしている。7, 8 行目はゼロ除算であるときのエラーの出力、9, 10 行目では演算子が適切でないときのエラーの出力を行っている。

11 行目以降では適切に数値と演算子が入力されたときに、それぞれの演算子の場合の計算結果を `answer` に格納している。

9 実行結果



```
tusedls00.ed.tus.ac.jp - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W)
tusedls05$ gcc io_report2.c
tusedls05$ a.out
+ 100
結果: 100.000000
- 50.3
結果: 49.700001
* 20
結果: 994.000000
/ 10
結果: 99.400002
? 10
エラー: 演算子ではない。
/ 0
エラー: ゼロ除算になる。
tusedls05$
```

図 2 実行結果 (2)

10 考察

このプログラムの工夫点は、演算子と数値を入力する際にそれぞれエンターを押したのちに入力しても、空白を開けて入力しても正常に認識される。プログラム上の `scanf("%")` 内で `%*c` と入力することで空白を認識できるようになるプログラムにした。