

1 Level3

任意の変数名、配列、Cの三項演算、while文、for文を実装せよ。

この簡単なコンパイラは、変数名は一文字しかみていない。これを辞書管理を行うことにより、任意の長さの変数名を可能にすることができる。

辞書管理部分を設計する必要がある。

項演算や、while文、for文などは、条件分岐を生成する必要がある。Micro-Cなども参考になるがあまりきれいなコードではない。以上をテストする入力を作成し、正しく動作することを調べよ。

辞書管理部分を設計するために、s-token.cのソースを、以下のように修正して、実行してみることにした。

(以下はs-token.cの修正した部分)

```
|
|
|
(省略)
|
|
|

} else if (('a'<=c && c<='z') ||
           ('A'<=c && c<='Z')) {
character[i]=c;
i++;
    while(c = *ptr++){
        if (('a'<=c && c<='z') ||
            ('A'<=c && c<='Z')) {
            character[i++] = c;
```

```
        }
        else{
            character[i++] = '\0';
            break;
        }
    }

    s=0;
    strcpy(data[0][s],character);
    value = s;
    c && ptr--;
    last_token = 'v';
    return last_token;
}
```

```
|
|
|
(省略)
|
|
|
```

(s-token.c の修正部分終了)

このプログラムのテストデータとして、s-input.txtの中身を

```
mouse=12
double=2*5
click=3+3
```

として、s-token.cを修正する前と修正後の実行結果を示す。

(以下、修正前実行結果)

```
[j04001@pw009 ~/examples]% make test
./s-intel-r < s-input.txt > s-output.s
syntax error on ouse=12

syntax error on ouble=2*5

syntax error on lick=3+3

syntax error on
syntax error on
gcc s-output.s
./a.out
[j04001@pw009 ~/examples]%
```

(修正前実行結果、終)

(以下、修正後実行結果)

```
[j04001@pw009 ~/unix2]% make test
./s-intel-r < s-input.txt > s-output.s
gcc s-output.s
./a.out
= 12
= 10
= 6
```