

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПРАВИЛНА ДУМА

Решение 1.

За решението на задачата е използвана функция `correct` (string `c`), която проверява дали думата е правилна по всички критерии зададени в условието на задачата.

С условен оператор проверяваме за нечетен брой букви в думата:

```
if (a[j].length()%2==0) return 0;
```

В цикъл превръщаме главна буква в съответната и малка, използвайки функцията `tolower`:

```
for (int j=0; j<c.length(); j++) c[j]=tolower(c[j]);
```

В цикъл правим проверка дали думата е правилна:

```
bool f=1;
```

```
for (int i=0; i<c.length(); i++) if (!c[i]-'0'>=c[i-1]-'0') {f=0; break;} return f;
```

В главната програма въвеждаме низа. Правим замяна на звездичките с интервал:

```
for (int i=0; i<s.length(); i++) if (s[i]=='*') s.replace (i, 1, " ");
```

и изтриваме символите:

```
for (int i=0; i<s.length(); i++)
```

```
if (s[i]=='%' || s[i]=='=' || s[i]=='+' || s[i]=='-' || s[i]=='(' || s[i]==')' || s[i]=='!' || s[i]=='.' || s[i]=='') {s.erase (i, 1); i--;}
```

Думите ще съхраним в масива `string a[128]={" "}`;

```
int br=0; - номера на текущата дума
```

Ако текущия символ е интервал и предходния не е интервал, означава, че сме свършили дума и увеличаваме индекса за записване на следващата дума. Ако не е интервал, означава, че имаме дума и добавяме текущия символ в записващата се дума в масива `a`:

```
for (int i=0; i<s.length(); i++) {if (s[i]==' ' && s[i-1]!=' ') br++; if (s[i]!=' ') a[br]+=s[i];}
```

```
int res_num, res_len=0; - номера и дължината на най-дългата дума
```

Обхождаме масива `a` по думи и извикваме функцията `correct` (string `c`) за да проверим дали думата е правилна. Правим проверка дали тази правилна дума е по-дълга от най-дългата правилна дума досега:

```
for(int j=0; j<br; j++)
```

```
if (correct(a[j]))
```

```
if (a[j].length()>res_len){ res_len=a[j].length(); res_num=j+1; }
```

Накрая извеждаме номера и дължината на най-дългата правилна дума:

```
cout<<res_num<<" "<<res_len<<endl;
```

Решение 2.

Задачата може да се реши и без да се преобразува низа, като вместо с интервали се работи със звездичките, а другите служебни символи се игнорират. Обхождайки низа при получаване на поредната дума, тя се проверява дали е правилна и съответно дали е по-дълга от най-дългата дума досега, без да се отделя в масив.

Кодът на това решение е в `word1.cpp`.

Автор: Мария Енева