

Решение:

1. Необходими величини:

Две стрингови променливи, съответно за думата и текста.

```
char d[20], s[1000];
```

Пет целочислени променливи, съответно за дължината на думата и текста, началото и дължината на най-дългата последователност от знаци на думата, които се съдържат в текста, както и за началото и дължината на текущата последователност. Началната стойност на всяка една от тези променливи трябва да бъде 0.

```
int n, mn=0, md=0, tn=0, td=0, i, j;
```

Допълнително се декларираят няколко работни променливи, които ще се използват за управление на циклите.

За определяне на принадлежността на даден знак към думата се използва помощен масив с 256 знака. Всеки елемент на този масив има стойност 0 ако знака със съответния код не е в думата и 1 – в противен случай. Този масив също се нулира в началото на програмата.

```
int b[256];  
for(i=0; i<256; i++) b[i] = 0;
```

2. Въвежда се думата, намира се нейната дължина и се запълва масива b.

```
cin >> d;  
n = strlen(d);  
for(i=0; i<n; i++) b[d[i]] = 1;
```

3. Въвежда се текста и се определя дължината му:

```
cin.getline(s, 1000, '\n');  
n = strlen(s);
```

4. Обхожда се текста и се намира най-дългата последователност от знаци на думата, които се съдържат в текста, като за всеки един от знаците на s, се проверява дали принадлежи на думата, т.е. дали съответното b е 1, при което се увеличава броя на буквите в текущата последователност от знаци, принадлежащи на думата. Ако знакът не принадлежи на думата са възможни два варианта: или това е края на текущата последователност и тогава трябва да проверим дали тя не е по-дълга от намерената до момента; или това е просто поредния знак от текста, който не принадлежи на думата и няма нужда от обработка.

```
for(i=0; i<n; i++)  
    if(b[s[i]]){ if(!td)tn = i; td++;}  
    else  
        if(td)  
        {  
            if(td > md){md = td; mn = tn;};  
            td=0;  
        }
```

5. За да не се пропусне случая когато най-дългата последователност остава последна в текста, обработката за най-дълга последователност се повтаря след като цикъла завърши:

```
if(td > md){md = td; mn = tn;};
```

6. Накрая се извежда началото и края на намерената най-дълга последователност.

```
cout << mn + 1 << ' ' << mn + md << endl;
```