プログラミング演習I　第6回　演習レポート

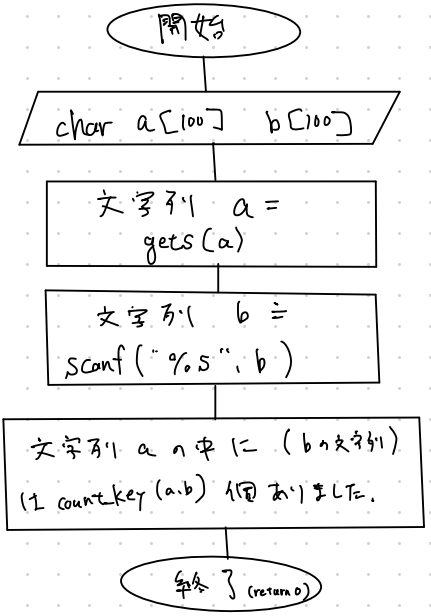
2164204　野村瑛吉

(1) 基本課題4

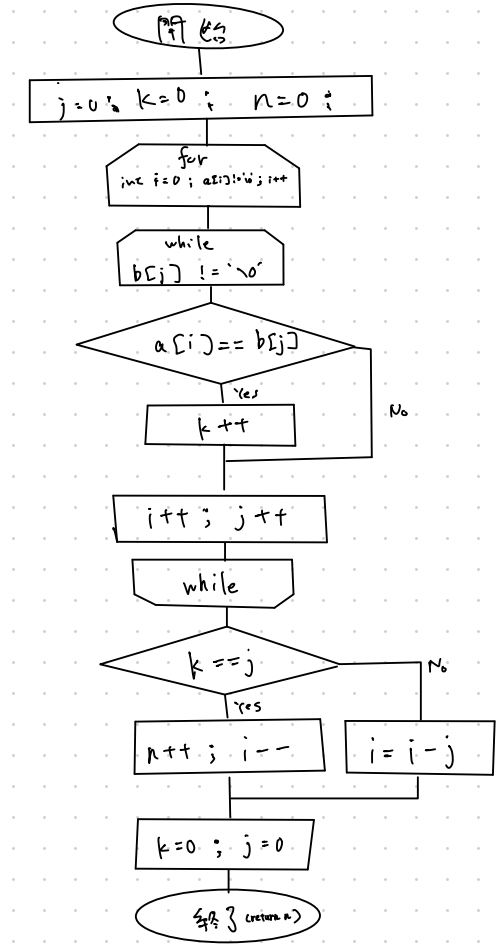
2164204_b4.c

指定したキーワードが含まれている個数を出力

(2) int main



int count_key



(3) main関数で文字数100の文字列
char a[100]、b[100]を定義.

aを探される側の文字列、bを探す文字列.
つまりaの中にbがいくつあるか数えるようにする.
文字列aをgets(a)で入力、文字列bをscanf関数で入力.

文字列aの中に文字列bがいくつ含まれているか
表す. この際にcount_key(a,b)が呼び出される.
int型関数count_key(char a[], char b[])では、
第一引数が探される側の文字列、
第二引数が探す側の文字列とし、
main関数内の文字列a[100]、b[100]と対応している.
int型変数j=0、k=0、n=0を定義.
for文では、文字列aの最後の文字までソート.
while文では、文字列bの最後まで、a[i]と同じ
文字がないか調べる. 同じ文字だった場合、kをインクリメントし、
同じ文字の場合もそうでない場合も、iとjをインクリメントする.
つまり、a[]もb[]も次の文字となり、b[j]='\0'となるまで
繰り返す.

for文の繰り返し i 回目において、

(*) ⎧ a[i] = b[0]　　(k=1)
　 ⎪ a[i+1] = b[1]　　(k=2)
　 ⎪ ⋮　　⋮
　 ⎩ a[i+j] = b[j-1]　　(k=j)　→ n++

のすべてが成立するとき、すなわち k=j となった場合、a[i] から a[i+j] が文字列 b[] と一致することになる。

このとき、a の中の b のカウント用変数 n の値を 1 増やすと数えることができる。

(*) がすべて成立しなかった、すなわち k<j の場合には、i → i+j となっているので、i → i-j (i+j → i) と直し、元の for 文に戻るようにする。

こうすることで、a 内のそれぞれの文字が最初となる b が存在するか調べることができる。

最終的な n の値が求める個数なので、これを戻り値とする。

a[ ] = "a_0 a_1 a_2 … \0" (max size 100)

b[ ] = "b_0 b_1 b_2 … \0" (max size 100)

---

i=0 a_0 a_1 a_2 … a_j ? = b_0 b_1 b_2 … b_j

i=1 a_1 a_2 a_3 … a_{1+j} ? = b_0 b_1 b_2 … b_j

⋮

(i=0) j=0 a_0 ? = b_0 (T) k++

j=1 a_1 ? = b_1 (T) k++

⋮ ⋮

a_j ? = b_j (T) k++

(不) j−− で重複てなくす

if k==j → n++, i−− else i=i−j (→ k=0, j=0)

(i=1) j=0 a_1 ? = b_0 (T) k++

a_2 ? = b_1 (T) k++

⋮ ⋮

a_{1+j} ? = b_j (T) k++

if k==j → n++, i−− else i=i−j (→ k=0, j=0)

---

k==j (文字列 a の中に文字列 b が存在)

の回数

= n ← 答え

(4)

main 関数で文字数 100 の文字列
char a[100]、b[100] を定義.

a を探される側の文字列、b を探す文字列.
つまり a の中に b がいくつあるか数えるようにする.

文字列 a を gets(a) で入力、文字列 b を scanf 関数で入力.

文字列 a の中に文字列 b がいくつ含まれているか
表す. この際に count_key(a,b) が呼び出される.

count_key 関数では、a[] に対する繰り返し
変数を i、b[] に対する繰り返し変数を j.
while 文内の if 文の True 数に対する変数を k、
文字列 a 内の文字列 b の個数を n とする。
for 文で文字列 a 内のすべての文字を調べる.
while 文で 文字列 b 内のすべての文字に対して
繰り返し、 a[i]==b[j] を i も j も
インクリメントしながら調べ、True 数を k とする.
if (k==j) は while 文で k が毎回インクリメント、
すなわち文字列 a の中に文字列 b が存在した場合に
実行され、n をインクリメントし、i をデクリメントすると

で重複を防ぐ。
if文がfalseだった場合、else文内の
i=i-jによって、while文でiがインクリメント
された分を元に戻す。
戻り値nが文字列aの中にある
文字列bの数となり、これをmain関数
で表示する。

(5) 今回のプログラムでは、for文とwhile文の
2重ループになっており、変数の扱いが
複雑しかった。今文字列aはどこの文字で
文字列bはどこの文字なのかしっかり
認識できていないといけなかった。
レポートによって、理解がとても深まった。
普段も文字で検索することは多々あるので、
そのアルゴリズムを体感できたのは、
とても興味深い。

(6) 一緒に考え、たくさんのヒントをくれた友人がいるので感謝する。

新村 将司 (B2、情報工学EP)

(7) 課題提出期限後、復習用の解答プログラムをLMSに公開して欲しい。