ProgramowanieProceduralne

IS_L7

`char* strtok(char* str, const char* set);`
funkcja ta jest „tokenizerem", czyli funkcją pozwalającą rozłożyć napis na leksemy („słowa") oddzielone znakami ze zbioru znaków w napisie set, które wobec tego traktowane są jako separatory (np. odstęp, przecinek, dwukropek, itd.).

Działanie funkcji oparte jest na istnieniu wewnętrznego wskaźnika typu **`char*`**. Kolejne wywołania **`strtok`** zwracają kolejne leksemy.

Przy pierwszym wywołaniu **`str`** jest napisem, który chcemy rozłożyć; w kolejnych wywołaniach pierwszym argumentem powinien być wskaźnik zerowy **`nullptr`** , co informuje funkcję, że należy kontynuować przetwarzanie od końca ostatnio zwróconego leksemu.
Pomiędzy wywołaniami nie wolno zmieniać napisu **`str`** □ sama funkcja go jednak zmienia, co oznacza, że nie można przekazać napisu niemodyfikowalnego, np. zadanego jako literał napisowy. Wolno natomiast przy kolejnych wywołaniach zmieniać zbiór separatorów w **`set`**.

```
#include <stdio.h>
#include <string.h>


  int main() {
  char strin[] = "int* fun(char& c,double** wtab);";
  char separ[] = ")(,;";
  char* token;

  token = strtok(strin,separ);
  while (token != 0) {
   printf ("%s \n",token);
   token = strtok(0,separ);
  }
}
```

`char* strcat(char* dest, const char* src);`
dodaje zawartość napisu **`src`** do napisu **`dest`** (konkatenacja); zwraca **`dest`**.

Użytkownik musi zadbać o to, by **`dest`** i **`src`** kończyły się znakiem **`'\0'`** i żeby obszar pamięci zarezerwowany pod adresem wskazywanym przez **`dest`** był wystarczający do pomieszczenia obu napisów (i kończącego znaku **`'\0'`**).

Pierwszy znak **`'\0'`** w napisie **`dest`** jest nadpisywany przez pierwszy znak **`src`** i poczynając od tej pozycji przekopiowywane są znaki z **`src`** aż do kończącego go znaku **`'\0'`** włącznie.

```
#include <stdio.h>
#include <string.h>
 int main(){
char tab[25] = "Ala ma ";
char tab2[] = "kota i psa";
printf("%s",strcat(tab,tab2));}
}
```

`char* strstr(char* src, const char* sub);`
zwraca wskaźnik do pierwszego znaku w **`src`**, od którego rozpoczyna się w nim podciąg identyczny z **`sub`** (pierwsze jego wystąpienie). Funkcja zwraca wskaźnik zerowy, jeśli podciąg **`sub`** nie występuje w **`src`**.

```
#include <stdio.h>
#include <string.h>
 int main(){
char tab[25] = "Ala ma kota, kotka i koteczka ";
char tab2[] = "kot";
printf("%s",strstr(tab,tab2));}
}
```

1.

(2) Proszę uzupełnić program sortujący tablicę stringów - dobrze jest zmieścić się w zaproponowanym schemacie.

```
#include <stdio.h>
#include <string.h>
#define ROZMIAR 81          //maksymalna długość łańcuchów
#define GRAN 20             // maksymalna liczba łańcuchów
void sortlan(char *lan[], int num);   //procedura sortująca łancuchy
int main(void)
{
  char dane[GRAN][ROZMIAR];     //tablica przechowujaca łańcuchy nie dłuzsze niz ROZMIAR znakow nie wiecej niz GRAN
  char **wsklan = ;     //wskaźnik na tablicę wskaźników - rozmiar zależny od ilości wczytanych łańcuchów
  int licz = 0;                 //licznik danych wejsciowych
  int k;                          */
  printf("Podaj maksymalnie %d wierszy \n",GRAN);
  printf("Aby zakonczyc, wcisnij Enter na poczatku wiersza.\n");

 while (licz < GRAN && gets(dane[licz]) != NULL && dane[licz][0] != '\0')
  {
     ......................................;
     ................. = ..................;     /ustaw wskaźnik z tablicy wsklan na wczytany łańcuch
     licz++;
  }
```

```
  puts("\n Wczytano:\n");
  for (k = 0; k < licz; k++)
     puts(.....................) ;    // z użyciem identyfikataora wsklan
```

```
sortlan(..............., .......); // wywłoanie procedury sortującej

 puts("\n Oto uporzadkowana lista:\n");
   for (k = 0; k < licz; k++) puts(.....................) ;  // z użyciem identyfikataora dane
```

```
puts("\n Oto uporzadkowana lista:\n");
   for (k = 0; k < licz; k++) puts(.....................) ;  // z użyciem identyfikataora wsklan
```

```
return 0; }

// procedura sortująca
void sortlan(char *lan[], int num) {
 //potrzebne zmienne lokalne
 for (.......; ........;.........)
 for (...............; ...............; .............)
 if (strcmp(.................,.....................) > 0)
 { ......................;
 ......................;
 ......................; }
}
```

2.

(3) Proszę napisać program, który przyjmuje jako **argument wiersza poleceń** łańcuch znaków.

Program czyta do końca plik lokomotywa.dat - przekierowanie strumienia wejściowego:

```
a.out tak < lokomotywa.dat
```

natomiast wyświetla tylko te linie wczytanego tekstu, które zawierają słowo podane jako **argument wywołania programu** .

Prosze podać ilość wystąpień tego słowa we wczytanym tekście - można przetestować dla słowa "tak" `wynik: wypisane 7 linii ; 12 wystąpień.`

3.

(6) Proszę napisać program, który przyjmuje jako **argument wiersza poleceń** łańcuch znaków. Program ma wyodrębnić wyrazy występujące w łańcuchu i stworzyć nowy łańcuch, który składa się z niepowtarzających się słów posortowanych alfabetycznie.
Wyrazy w wejściowym łańcuchu mogą być rozdzielone następującymi znakami `',' ' ' ';' '.' ':'`.
W programie należy stworzyć i wykorzystać następujące funkcje:

- funkcja tworząca tablicę niepowtarzających się słów z łańcucha
  ```
  char** podziel(char *, int *);
  ```
- funkcja sortująca tablicę słów
  ```
  void sort(char **, int);
  ```
- funkcja „sklejająca” słowa z tablicy (rozdzielane spacjami) w JEDEN łańcuch
  ```
  char* sklej(char **, int);
  ```

Przykład:
./a.out "The sun did not shine. It was too wet to play. So we sat in the house - All that cold, cold, wet day."
All It So The cold day did house in not play sat shine sun that the to too was we wet

---

4.
(2) Proszę zdefiniować funkcję, która zwraca różnicę dwóch liczb całkowitych, oraz funkcję, która zwraca sumę liczb całkowitych.
Niech prototypy funkcji mają postać:

```
int add2(int * , int *);
int comp2(int * , int *);
```

Proszę zdefiniować funkcję

```
int add2_or_comp2 (............., ............., .............);
```

której argumenty to:

- dwie wartości całkowite,
- wskaźnik do funkcji, która pobiera dwa wskaźniki na `int`, a zwraca wartość typu `int`

funkcja `add2_or_comp2()` ma zwracać wartość typu `int`, która jest wynikiem przekazanej przez wskaźnik funkcji.

W funkcji `main()` proszę wczytać dwie wartości całkowite, a następnie korzystając z funkcji `add2_or_comp2()` wypisać ich sumę oraz różnicę .

Status przesłanego zadania

Status przesłanego zadania	Przesłane do oceny
Stan oceniania	Nieocenione
Termin oddania	poniedziałek, 6 kwietnia 2020, 14:25
Pozostały czas	11 sek.
Ostatnio modyfikowane	poniedziałek, 6 kwietnia 2020, 14:22
Przesyłane pliki	1.c — 6 kwietnia 2020, 13:40 2.c — 6 kwietnia 2020, 13:45 3.c — 6 kwietnia 2020, 14:22 4.c — 6 kwietnia 2020, 13:45
Komentarz do przesłanego zadania	Komentarze (1)

Edytuj zadanie | Usuń zadanie

Możesz nadal zmieniać złożone zadanie.

◄ LAB_7

Przejdź do...

Platforma e-Learningowa obsługiwana jest przez:
Centrum e-Learningu AGH oraz Uczelniane Centrum Informatyki AGH

Podsumowanie zasad przechowywania danych
Pobierz aplikację mobilną