Zadanie D
Napisy

Język programowania: C++
Autor: Daniel Wilczak

Państwa zadaniem jest przygotowanie zestawu funkcji wykonujących pewne operacje na napisach. Nazwy funkcji oraz ich argumenty zostaną opisane poniżej.

Sposób przesyłania rozwiązania

Do systemu BaCa należy przesłać plik o nazwie source.cpp zawierający definicje wszystkich wymaganych w zadaniu funkcji, ale nie zawierający funkcji main. Programy testujące Państwa rozwiązanie będą miały następującą strukturę:

```
#include <cstdarg>
#include "source.cpp"

// funkcje pomocnicze testu

int main(){
  // wlaściwy test rozwiązania
}
```

WAŻNE:

1.Rozwiązanie nie może wykorzystywać **żadnych** funkcji bibliotecznych z wyjątkiem tych zdefiniowanych w **cstdarg**. Dlatego nie są dozwolone żadne dyrektywy preprocesora. Jeżeli w pliku z rozwiązaniem zostanie znaleziony znak # (również w komentarzu), to zostanie zgłoszony **błąd kompilacji programu**.
2.W rozwiązaniu nie wolno użyć **żadnego** ze słów kluczowych **int, long, unsigned, template, typedef** oraz nawiasów **[]**. Słowa te są zabronione tak w kodzie, jak i w komentarzach.
3.Jeśli przy opisie funkcji nie podano jaką ma zwracać wartość, to znaczy, że nie jest to istotne i nie będzie sprawdzane.
4.Funkcje powinny być odporne na błędy danych wejściowych (w slangu informatycznym "idiotoodporne"). Jeżeli przy opisie funkcji nie jest podane, że argumenty będą poprawne, to należy obsłużyć wszystkie możliwe scenariusze. Nie precyzuję tych scenariuszy, aby dać Państwu szansę wcielenia się w rolę testera oprogramowania.

Lista funkcji, które należy napisać.

Poniżej podana jest lista funkcji, które należy napisać. Nadesłane rozwiązanie może oczywiście zawierać dowolną liczbę funkcji pomocniczych.

1.funkcja o nazwie **przeksztalc**. Jej argumentami są:
- •napis **s** (tablica znakowa zakończona zerem),
- •niezerowy wskaźnik do funkcji **f** o jednym argumencie typu **char** i wartości typu **char**.

Zadaniem funkcji **przeksztalc** jest zamiana każdego znaku napisu **s** na wartość funkcji **f** obliczonej dla tego znaku.

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>

char zmienNaWielkaLitere(char c){
  return (c>='a' and c<='z') ? c-32 : c;
}

int main(){
  char s[] = "Ala ma kota";
  przeksztalc(s,zmienNaWielkaLitere);
  printf("%s\n",s);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć napis:

```
ALA MA KOTA
```

2.funkcja o nazwie **filtruj**. Jej argumentami są:
- •napis **s** (tablica znakowa zakończona zerem),
- •niezerowy wskaźnik do funkcji **f** o jednym argumencie typu **char** i wartości typu **bool**.

Zadaniem funkcji **filtruj** jest usunięcie z napisu tych znaków, dla których funkcja **f** zwraca **true**. Usunięcie znaku oznacza, że następujące po nim znaki są przesuwane w lewo o jedną pozycję.

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>
```

```
bool zostawMalaLitere(char c){
  return (c<'a' or c>'z');
}

int main(){
  char s[] = "Ala ma kota";
  filtruj(s,zostawMalaLitere);
  printf("%s\n",s);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć napis:

```
lamakota
```

3.funkcja o nazwie **filtruj**. Jej argumentami są:

- napis **s** (tablica znakowa zakończona zerem),
- napis **k** (tablica znakowa zakończona zerem).

Zadaniem funkcji **filtruj** jest usunięcie z napisu **s** wszystkich znaków, które występują w napisie **k**.

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>

int main(){
  char s[] = "Ala ma kota";
  filtruj(s,"ao");
  printf("%s\n",s);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć napis:

```
Al m kt
```

4.funkcja o nazwie **filtruj**. Jej argumentami są:

- tablica napisów podana jako poprawny zakres **[b,e)**, gdzie **b** jest adresem pierwszego napisu, natomiast **e** jest adresem "następnego napisu za ostatnim"
- wskaźnik do funkcji **f** o jednym argumencie typu **char** i wartości typu **bool**.

Zadaniem tej wersji funkcji **filtruj** jest zastosowanie wcześniejszej funkcji **filtruj** do każdego napisu z zakresu **[b,e)**. Funkcja ma zwrócić wartość logiczną:

•**true** - w przypadku, gdy którykolwiek z napisów z tablicy napisów **[b,e)** został zmodyfikowany przez **filtruj**

•**false** - w przypadku, gdy żaden z napisów z tablicy napisów **[b,e)** nie został zmodyfikowany przez **filtruj**

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>

bool zostawMalaLitere(char c){
  return (c<'a' or c>'z');
}

int main(){
  char s1[] = "Ala ma kota!";
  char s2[] = "Kot ma Ale?";
  char* s[] = {s1,s2};
  int d = filtruj(s,s+2,zostawMalaLitere);
  printf("%s, %s, %d\n",s[0],s[1],d);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć:

```
lamakota, otmale, 1
```

5.funkcja o nazwie **szyfruj** i argumentach

•napis **s** (tablica znakowa zakończona zerem) do zaszyfrowania,

•klucz szyfru **klucz** (również napis)

Zadaniem tej funkcji jest zaszyfrowanie napisu szyfrem Vigenère'a (zobacz opis na stronach Wikipedii:

```
https://pl.wikipedia.org/wiki/Szyfr_Vigen%C3%A8re%E2%80%99a
```

Uwagi:

•szyfrujemy jedynie litery alfabetu angielskiego (małe litery przechodzą na małe, a wielkie na wielkie),

•znaki, które nie są literami alfabetu angielskiego pozostawiamy bez zmian; nie zmieniamy również pozycji w kluczu,

•należy przyjąć, że klucz składa się jedynie z wielkich liter alfabetu angielskiego i ma co najmniej jeden znak

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>
```

```
int main(){
  char s1[] = "TO JEST BARDZO TAJNY TEKST!@#$%";
  char s2[] = "to jest bardzo tajny tekst!@#$%";
  char klucz[] = "TAJNE";
  szyfruj(s1,klucz);
  szyfruj(s2,klucz);
  printf("%s\n%s",s1,s2);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć:

```
MO SRWM BJEHSO CNNGY CROLT!@#$%
mo srwm bjehso cnngy crolt!@#$%
```

6.funkcja o nazwie **przetwarzaj** o nieokreślonej liczbie argumentów. Jej pierwszym i jedynym określonym argumentem jest klucz **k** jako tablica znakowa dowolnej długości zakończona zerem i składająca się jedynie z liter **f,p,s**. Kolejne argumenty wywołania są zdeterminowane przez liczbę i wartość liter w kluczu **k** według następujących reguł:

- **znak f** oznacza filtrowanie. Kolejnymi argumentami wywołania funkcji będą: napis oraz funkcja filtrująca (o argumencie **char** i wartości **bool**). Po napotkaniu litery **f** w kluczu należy wywołać funkcję **filtruj** z tymi argumentami.
- **znak p** oznacza przekształcenie. Kolejnymi argumentami wywołania funkcji będą: napis oraz funkcja przekształcająca (o argumencie **char** i wartości **char**). Po napotkaniu litery **p** w kluczu należy wywołać funkcję **przeksztalc** z tymi argumentami.
- **znak s** oznacza szyfrowanie. Kolejnymi argumentami wywołania funkcji będą dwa napisy, powiedzmy **napis,kod**, z których pierwszy jest napisem do zaszyfrowania, a drugi jest kluczem szyfru. Po napotkaniu litery **s** należy wykonać operację szyfrowania napisu **napis** kluczem **kod**

Uwaga: Należy przyjąć, że funkcja będzie wywoływana z poprawną listą argumentów nieokreślonych. Czyli dla każdego znaku **f,p,s** klucza będą wymagane argumenty wywołania.

Test jawny:

```
#include <cstdarg>
#include "source.cpp"
#include <cstdio>

char zmienNaWielkaLitere(char c){
  return (c>='a' and c<='z') ? c-32 : c;
}
bool zostawMalaLitere(char c){
  return (c<'a' or c>'z');
```

```
}

int main(){
  char s1[] = "Ala ma kota.";
  char s2[] = "Kot ma Ale!";
  char s3[] = "to jest bardzo tajny tekst";

przetwarzaj("pfs",s1,zmienNaWielkaLitere,s2,zostawMalaLitere,s
3,"TAJNE");
  printf("%s\n%s\n%s\n",s1,s2,s3);
  return 0;
}
```

Po wykonaniu tego programu na standardowym wyjściu powinniśmy zobaczyć:

```
ALA MA KOTA.
otmale
mo srwm bjehso cnngy crolt
```