

Zadanie I - Podprogramy

Punktów procentowych do uzyskania: **10**

Język programowania: C++

autor zadania: Rafał Kawa

- System BaCa zawiera kod źródłowy programu zadania I z wypełnionym ciałem, ale z brakującą implementacją użytych podprogramów będących celem zadania.
- Ciało głównego programu dla uzyskaniu wyników stosuje podprogramy o identyfikatorach `Sum` oraz `Mult` implementujące odpowiednio sumowania oraz mnożenia dowolnie dużych liczb całkowitych w dowolnej dużej ilości (jednakże nie mniejszej niż dwa), z argumentami jak i zwracanymi wartościami przekazywanymi w formie ciągów znakowych.
- Ponadto główny program może korzystać z podprogramów o identyfikatorach `Operation` implementujących możliwość zastosowania w pojedynczym wywołaniu jednej z powyższych funkcji przekazywanych argumentem wraz z pozostałymi argumentami przekazującymi argumenty dla wywołań przekazywanych podprogramów.
- Stosowane w głównym programie podprogramy spełniają wyłącznie jeden z poniższych wzorców nagłówkowych:

```
string Sum ( int, const string* );  
string Sum ( int, ... );  
void Sum ( string*, int, const string* );  
void Sum ( string*, int, ... );  
void Sum ( string&, int, const string* );  
void Sum ( string&, int, ... );  
string Mult ( int, const string* );  
string Mult ( int, ... );  
void Mult ( string*, int, const string* );  
void Mult ( string*, int, ... );  
void Mult ( string&, int, const string* );  
void Mult ( string&, int, ... );  
string Operation ( string (*) ( int, const string* ), int, const string* );  
string Operation ( string (*) ( int, const string* ), int, ... );  
void Operation ( string*, string (*) ( int, const string* ), int, ... );  
void operation ( string*, string (*) ( int, const string* ), int, const string* );  
void operation ( string&, void (*) ( string*, int, const string* ), int, const string* );  
void operation ( string&, void (*) ( string*, int, const string* ), int, ... );
```

gdzie:

- Wartości zwracane przez podprogramy zawsze służą przekazaniu wyników działań.
- Argumenty całkowite zawsze oznaczają ilość argumentów podlegających działaniu.
- Argumenty wskaźnikowe lub referencyjne o ile są pierwszymi argumentami zawsze służą zwróceniu efektów działań.
- Argumenty wskaźnikowe o ile są ostatnimi służą przekazaniu liczb podlegających działaniu.
- Argumenty adresowe podprogramów odnoszą się do istniejących podprogramów `Sum` lub `Mult`.
- Kody wszystkich podprogramów możliwych do zastosowania w ciele programu wraz z ewentualnymi własnymi podprogramami muszą być zawarte w pliku o nazwie `subprograms.cpp` z uwzględnieniem faktu, iż ogólne ramy kodu mają postać:

```
#include <cstdlib>  
#include <iostream>  
#include <string>  
using namespace std;  
#include "subprograms.cpp"  
int main ()  
{  
    // istniejące ciało programu  
    return 0;  
}
```

- W przekazywanym na BaCę kodzie źródłowym nie jest możliwe stosowanie jakichkolwiek plików nagłówkowych.
- Na BaCę wysyłany jest tylko i wyłącznie SPAKOWANY programem ZIP plik `subprograms.cpp`.