LABORATORIUM PODSTAW PROGRAMOWANIA

LAB 3 FUNKCJE

Każdy program napisany w języku C/C++ zawiera przynajmniej jedną funkcję o predefiniowanej nazwie: **main( )**. Najczęściej wykorzystuje się również wiele innych predefiniowanych funkcji np. **printf(...)**, **scanf(...)**, **abs(...)**, **sin(...)**, itp. Można również definiować nowe-własne funkcje.

Składnia definicji funkcji:

```
zwracany_typ NAZWA_FUNKCJI ( lista parametrów )
{
 instrukcja lub sekwencja instrukcji ;

}
```

przykład:

```
int MAX ( int liczba_1 , int liczba_2 )
{
if( liczba_1 > liczba_2 ) return liczba_1 ;
 else return liczba_2 ;
}
```

- lista parametrów może być pusta lub zawierać opisy kolejnych parametrów (pooddzielane przecinkami):
  `main( ) main( void ) main( int argc , char* argv[ ] )`

- parametry definiowane są tak jak zmienne, nie można grupować sekwencji parametrów tego samego typu:
  ~~`int MAX ( int liczba_1, liczba_2, liczba_3 )`~~ źle !
- „ciało" funkcji jest zawarte pomiędzy nawiasami: **{ ... }** (bez średnika na końcu)
- działanie funkcji kończy się po napotkaniu polecenia **return** lub po wykonaniu sekwencji wszystkich instrukcji zawartych w ciele funkcji,
- jeżeli funkcja jest typu **void**, to używamy samego słowa **return**, bez żadnego wyrażenia po nim,
- jeżeli funkcja jest typu innego niż **void** to po poleceniu return **musi** się pojawić wyrażenie odpowiedniego typu (może być w nawiasach), np.:

  `return liczba_1; lub return( liczba_1 ) ;`

Prototyp funkcji - deklaracja „uprzedzająca", określa tylko nazwę funkcji oraz typ zwracanej wartości i parametrów (sam nagłówek funkcji zakończony średnikiem) Taka deklaracja funkcji jest konieczna w przypadkach, gdy wywołanie funkcji występuje wcześniej niż jej definicja. Np.

```
#include <….>
int mojaF(int,char);
int main(void)
```

```
{
 int a,w;
 char b;
 …
 w=mojaF(a,b);
 …
}

int mojaF(int w1,char w2)
{
 int z;
 ….
 return z;
}
```

W języku C parametry mogą być przekazywane na dwa sposoby : przez wartość i przez adres. Jeżeli argument przekazywany jest przez wartość, to po wywołaniu funkcji tworzone są nowe zmienne (lokalne), których zawartość inicjowana jest wartościami parametrów zmiennych, stałych lub wyrażeń) podanych przy wywołaniu. Funkcja może zmieniać wartości tych zmiennych, ale zmiany nie są widoczne na zewnątrz.

Zadania

1) Napisz program, który będzie wyświetlał na ekranie postać binarną liczby dziesiętnej podanej przez użytkownika. Zakładamy, że zmienna, w której będzie przechowywana ta liczba jest typu integer. Program należy zbudować z fukcji. Zadaniem pierwszej z nich będzie komunikacja z użytkownikiem (wraz ze sprawdzeniem poprawności wprowadzonej przez niego wartości), a drugiej znalezienie reprezentacji binarnej tej wartości. Uwaga, funkcje nie mogą bezpośrednio odwoływać się do zmiennych globalnych.
2) Zmodyfikuj program z pkt 1 tak aby zrealizować prosty kalkulator realizujący operacje bitowe and, or, xor, negacja. Program pobiera parametry bezpośrednio z linii komend (parametry wywołania programu), są to 2 liczby całkowite i operator, wyświetla argumenty oraz wynik w postaci binarnej
3) Zdefiniuj funkcję silnia, która dla danej liczby naturalnej oblicza wartość n!, przeanalizuj wpływ typu zmiennych na otrzymany wynik (kilka wersji funkcji), napisz drugą wersję funkcji silnia w wersji rekurencyjnej. Funkcje umieść w osobnym pliku źródłowym, wynik obliczania przypisz do zmiennej globalnej również zdefiniowanej w osobnym pliku źródłowym.