Funkcje o zmiennej liczbie argumentów

Język C pozwala definiować funkcje o zmiennej liczbie argumentów (ang. variadic functions). Oznacza to, że liczba argumentów, które możemy przekazać do takiej funkcji ograniczona jest jedynie rozmiarem stosu. Na liście argumentów formalnych funkcji o zmiennej liczbie argumentów powinien znaleźć się choć jeden „zwykły" argument, który nie jest zmienną rejestrową ani wskaźnikiem. Jeśli takich argumentów będzie na liście więcej, to ta uwaga dotyczy ostatniego z nich. Po zwykłych argumentach w liście argumentów funkcji o zmiennej liczbie argumentów występuje przecinek i trzy kropki (...).

Aby obsłużyć dodatkowe argumenty wywołania potrzebny jest typ ***va_list*** i odpowiednie makra zdefiniowane w pliku nagłówkowym ***stdarg.h.***

Makro ***va_start*** przyjmuje dwa argumenty, pierwszym jest zmienna typu ***va_list***, którą to makro inicjalizuje, a drugim nazwa ostatniego argumentu spośród „zwykłych" argumentów umieszczonych na liście argumentów funkcji. To makro musi zostać wywołane przed wszystkimi innymi, które zostaną opisane. Zmienna typu ***va_list*** jest listą wszystkich argumentów dodatkowych, przekazanych do funkcji.

Makro ***va_arg*** umożliwia poruszanie się po tej liście. Przyjmuje ono dwa parametry. Pierwszym jest lista parametrów dodatkowych, drugim typ wartości, która zostanie zwrócona.

Wywoływanie makra ***va_arg*** umieszczone jest najczęściej w pętli. Przy każdej iteracji pętli makro to zwraca wartość bieżącego argumentu na liście, przyjmując typ wartości tego argumentu taki, jaki został określony drugim argumentem wywołania marka oraz modyfikuje tak listę, aby po jego ponownym wywołaniu zwrócić następny element z tej listy. Po przejrzeniu całej listy należy wywołać ***makro va_end***, które jako argument wywołania przyjmuje listę argumentów dodatkowych. Jeśli ponownie chcielibyśmy przejrzeć tę listę, to musimy jeszcze raz użyć wyżej wymienionych makr. Ostatnim makrem związanym z obsługą listy argumentów dodatkowych jest ***va_ copy***. Służy ono, zgodnie ze swoją nazwą do tworzenia kopii listy argumentów. Nie należy do tego celu używać instrukcji przypisania!

W przykładzie poniżej przedstawiono program z funkcją, która liczy średnią wartości jej argumentów dodatkowych. Przyjęto w niej, że pierwszy argument przechowuje liczbę argumentów dodatkowych. W wywołaniu takiej funkcji można przekazać jej argumenty, które są literałami (wartościami), wyrażeniami, stałymi i zmiennymi.

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

float average(int counter, ...)
{
        va_list ap;
        int next=0, sum=0, i=counter;

        if(counter==0)
                return 0.0;

        va_start(ap,counter);
        while(i--) {
                next = va_arg(ap,int);
                sum += next;
        }
        va_end(ap);
        return (float)sum/counter;
}


int main(void)
{
        float result = average(4,1,2,3,4);
        printf("%f\n",result);
        return 0;
}
```

W języku C prototyp funkcji zapisany np. tak: int funkcja() również oznacza funkcję, która

przyjmuje zmienną liczbę argumentów. Jeśli chcemy stworzyć funkcję, która nie przyjmuje w ogóle argumentów, to powinniśmy jej prototyp zapisać jako int funkcja(void).

Wykonaj funkcje dla różnych typów argumentów dodatkowych wprowadzająca na przemian cyklicznie char, int, float i wypisujacą wprowadzone argumenty.