

## ZADANIA Z JĘZYKA C DLA GRUP 7. I 9.

### Zestaw III - grudzień 2015

Uzupełnienie do Zadania 1. (o obliczaniu średniej arytmetycznej): W programie wczytującym dane ze standardowego wejścia można użyć konstrukcji

```
while (EOF!=scanf("%lf",&x)) { ... };
```

a wewnątrz pętli instrukcji `break` dla przypadku, gdyby wczytanie kolejnej liczby spowodowało przekroczenie założonego maksymalnego rozmiaru tablicy na dane (`nmax`). **Na wejście programu** można skierować, korzystając z mechanizmu potoków unixowych, plik zawierający kolumnę np. 1000 liczb pseudolosowych o rozkładzie równomiernym w przedziale  $0 \leq x < 1$  (wygenerowanych osobnym programem). W takim przypadku, otrzymane wyniki powinny być bliskie:  $\langle x \rangle \simeq 0.5$  (średnia),  $\sigma(x) \simeq \frac{1}{6}\sqrt{3} \approx 0.289$  (odchylenie standardowe).

11. **Konwersja liczby całkowitej do postaci dwójkowej.** Proszę napisać funkcje

```
int Konwersja_Dec_Bin(unsigned dec, char *bin);
```

która konwertuje liczbę całkowitą (`dec`) do postaci tablicy znakowej (`bin`) zawierającej ciąg cyfr 0 i 1 odpowiadający zapisowi tej liczby w systemie dwójkowym; wartość zwracana to liczba cyfr w zapisie dwójkowym (nie licząc wiodących zer). W programie wywołującym tę funkcję, zmienną `dec` wczytujemy z klawiatury zaś tablicę `bin` wypisujemy na ekran za pomocą funkcji biblioteki standardowej (`scanf` i `printf`). Rozmiar tablicy `bin` można powiązać ze stałą `WORD_BIT` zdefiniowaną w pliku `limits.h`. Zadanie należy wykonać na dwa sposoby:

- za pomocą dzielenia `%2` i standardowych operacji arytmetycznych (`+`, `-`, `*`, `/`);
- za pomocą operacji bitowych `x&01` oraz `x=x>>1` (ewentualnie w wersji z przesunięciem bitowym w lewo: `x&a` oraz `a=01<<i`, gdzie `i=0,1,2,...`).

12. **Mnożenie z przesunięciami bitowymi.** Proszę napisać funkcje, która mnoży dwie liczby typu `unsigned` (albo `unsigned long`), a wynik zapamiętuje w dwóch innych zmiennych, z których jedna przechowuje najmłodsze bity (ściśle mówiąc: wynik mnożenia modulo `UINT_MAX` albo `ULONG_MAX`), zaś druga - pozostałe starsze bity. Funkcje można następnie wykorzystać do implementacji, w języku ANSI C, generatora liczb pseudolosowych Parka-Millera, który tworzy ciąg liczb opisany wzorem

$$r_{n+1} = 48271 \cdot r_n \bmod (2^{31} - 1),$$

począwszy od pewnej początkowej wartości np.  $r_0 = 1$  (to tzw. *podstawa* generatora). Łatwo zauważyć, że próba implementacji takiego generatora z wykorzystaniem zwykłego mnożenia liczb całkowitych szybko doprowadzi do przepełnienia zakresu i nieprawidłowych wyników.