***Rekurencja***

W algorytmice mówimy, że dany algorytm jest rekurencyjny, jeśli do rozwiązania pewnego problemu wykorzystuje on sam siebie. W programowaniu dana funkcja jest rekurencyjna, jeśli wywołuje samą siebie. Ilość takich wywołań nie może być nieskończona ponieważ pamięć komputera jest skończona. Dlatego w rekurencji bardzo ważne jest określenie warunku, który kończy rekurencję - np. funkcja przestaje już dalej wywoływać samą siebie.

**Zad.1** Napisz wersję iteracyjną i rekurencyjną funkcji obliczania potęgi o wykładniku naturalnym. W main() wprowadź przykładowe dane i wywołaj te funkcje.

**Zad.2** Napisz wersję iteracyjną i rekurencyjną funkcji obliczania n-tego wyrazu ciągu Fibonacciego. W main() wprowadź przykładowe dane i wywołaj te funkcje.

**Zad.3** Napisz wersję iteracyjną i rekurencyjną funkcji obliczania silni. W main() wprowadź przykładowe dane i wywołaj te funkcje.

W rozwiązaniu powyższych zadań pomocny będzie: <http://miroslawzelent.pl/kurs-c++/rekurencja-rekursja/>

**Wieże Hanoi** – to typowy problem, do rozwiązania którego wykorzystuje się rekurencję.

***Legenda*** *-* Jak głosi stara hinduska legenda, przy stworzeniu świata, w jego środku, pod dachem świątyni, umieszczone zostały trzy diamentowe pałeczki. Na jedną z nich nałożone były 64 złote krążki o zmniejszających się średnicach tworząc złoty stożek. Dzień i noc, zmieniając się bezustannie, mnisi przekładali krążki na trzecią pałeczkę. Musieli jednak zachować pewne zasady. Mogli posiłkować się drugą pałeczką, jednakże nie wolno było im przenosić więcej niż jednego krążka i umieszczać większego na mniejszym. Gdy wykonają swoje zadanie - nastąpi koniec świata!

***Obliczenie daty końca świata*** *–* Przyjmując, że przełożenie 1 krążka trwa 1 sekundę. Przełożenie 64 krążków będzie trwało:

264−1 = 18 446 744 073 709 551 615 (blisko 18 i pół tryliona) sekund = około 584 542 mld lat

Wszechświat ma (tylko!) około 13,7 mld lat.

Jeśli masz troszkę mniej czasu to: <http://www.zagraj.republika.pl/hanoi.html>

Funkcja ”hanoi” pokazuje jak przekładać krążki aby ułożyć wieżę.

void hanoi ( int n , char a, char c, char b);

int main(int argc, char \*argv[])

{

cout <<"Ile krazkow jest na poczatkowym paliku?";

int ile;

cin>>ile;

hanoi (ile,'A','C','B');

}

//---------------------------

void hanoi ( int n , char a, char c, char b)

{

if (n==1)

cout << a << " -> " << c << endl;

else

{

hanoi (n-1,a,b,c);

cout << a << " -> " << c <<endl;

hanoi (n-1,b,c,a);

}

}