



LAB 3

Algorytmy i struktury danych – rand(),srand()
generator liczb pseudolosowych - zastosowania.

Generator liczb pseudolosowych rand/srand - w praktyce

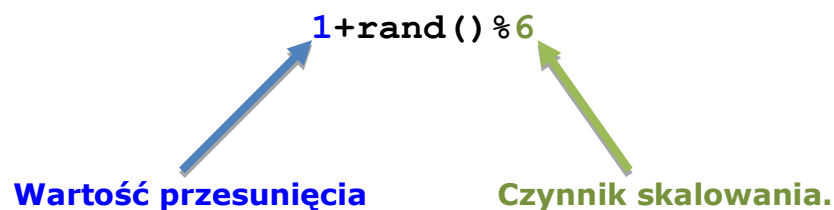
Do generowania liczb pseudolosowych służy funkcja **rand()** (ang. *Random* - losowy).

Funkcja ta generuje liczbę całkowitą między **0** i **RAND_MAX** (stałą symboliczną zdefiniowaną w pliku nagłówkowym cstdlib).

SKŁADNIA:

```
#include <cstdlib>
int rand();
```

W celu wygenerowania liczb np. od 1 do 6 używamy symbolu (%) w połączeniu z funkcją **rand**:

**Przykłady:**

```
Rand() % 100            -> zakres 0 do 99
Rand() % 100 + 1       -> zakres 1 do 100
Rand() % 30 + 1985     -> zakres 1985 do 2014
1 + Rand() % 100       -> zakres 1 do 100
1985 + Rand() % 30     -> zakres 1985 do 2014
```

Program symulujący rzut kostką 20 razy

Przykład 1

```
#include <stdio.h>
#include <stdlib.h>
using namespace std;
int main(int argc, char *argv[])
{
    int i=1;
    cout <<"=====\\n";
    cout <<"      Program symulujący rzut kostką 20 razy      \\n";
    cout <<"      Niestety losuje te same liczby      \\n";
    cout <<"      WWSSE(C)2019 student      \\n";
    cout <<"=====\\n\\n";

    for(; i<21; i++)
        cout<<(1+rand()%6)<<endl;
    cout<<endl;

    system("PAUSE");
    return 0;
}
```

W celu ustawienia pseudolosowego generatora numerów zanim zacznie się korzystać z funkcji **rand** należy użyć funkcji **srand()** w połączeniu z wyrażeniem **time**.

```
srand(time(0));
srand(time(NULL));
```

Funkcja **srand()** inicjuje generator liczb pseudolosowych podanym w argumencie zarodkiem (**ang. seed**). Użycie argumentu **time(0)** powoduje, że za każdym uruchomieniem programu zarodek liczb pseudolosowych jest inny☺.

Przykład 2

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h> //zwraca czas lokalny☺
using namespace std;
int main(int argc, char *argv[])
{
    int i=1;
    cout <<"=====\\n";
    cout <<"      Program symulujący rzut kostką 20 razy      \\n";
    cout <<"      Losuje różne liczby za każdym uruchomieniem      \\n";
    cout <<"      WWSSE(C)2019 student      \\n";
    cout <<"=====\\n\\n";
    srand(time(0));
    for(; i<21; i++)
        cout<<(1+rand()%6)<<endl;
    cout<<endl;

    system("PAUSE");
    return 0;
}
```

Instrukcja wielokrotnego wyboru switch

```
switch (warunek)
{
    case 1: instrukcja1; break;
    case 5: instrukcja2; break;
    case 8: instrukcja3; break;
    default: instrukcja4;
}
```

Gdy zmienna **warunek** będzie miała wartość **1**, wykonają się wszystkie instrukcje od dwukropka do najbliższej instrukcji **break** [czyli **instrukcja1**]. Gdy zmienna **warunek** będzie miała wartość **5**, wykona się **instrukcja2**. No i wreszcie gdy zmienna **warunek** będzie miała wartość **8**, wykona się **instrukcja3**. Jeśli zmienna **warunek** nie będzie równa żadnej z wymienionych wartości, wykonają się instrukcje "awaryjne" umieszczone po słówku **default**. Jeśli zapomnimy słówka **break**, program będzie wykonywał wszystkie instrukcje po kolei dopóki nie napotka instrukcji **break** albo dopóki nie skończy się instrukcja warunkowa. Zazwyczaj jest to efekt niepożądany.

Przykład 1

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main(int argc, char *argv[])
{
    int boka, wybor, pole, obwod;

    cout << "Program oblicza pole i obwod kwadratu\n";
    cout << "Wybierz co chcesz obliczyc:\n";
    cout << "1 -> Pole\n";
    cout << "2 -> Obwod\n";
    cin >> wybor;
    switch (wybor)
    {
        case 1:
            cout << "Podaj dlugosc boku kwadratu a="; cin>>boka;
            pole=boka*boka;
            cout << "Pole kwadratu wynosi: "<<pole<<endl;
            break;
        case 2:
            cout << "Podaj dlugosc boku kwadratu a="; cin>>boka;
            obwod=4*boka;
            cout << "Obwod kwadratu wynosi: "<<obwod<<endl;
            break;
        default:
            cout << "Prosze wybrac jedna z opcji 1 lub 2";
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

ZADANIA

1. Napisz program, pobierający z klawiatury liczbę całkowitą i mówiącą o tym czy jest ona z przedziału $<a;b>$.
2. Napisz algorytm wyświetlający tablicę kodów i znaków ASCII za pomocą pętli for. Wyświetl numer i odpowiadający mu znak.
3. Napisz algorytm losujący:
 - a) n liczb w wykorzystaniem generatora liczb pseudolosowych. Wartość n podawana jest przez użytkownika i jest liczbą naturalną.
 - b) n liczb o określonym zakresie losowania $od - do$ z wykorzystaniem generatora liczb pseudolosowych. Wartość n, od, do podawane są przez użytkownika i liczbami naturalnymi,
4. Napisz grę „**Kości**”, w której komputer „rzuca kostką”, a następnie pyta użytkownika o wylosowaną liczbę oczek.

Założenia:

- Domyślnie komputer rzuca jedną kością.
 - Jeśli użytkownik poda prawidłową liczbę oczek zostanie ona wyświetlona na ekranie z informacją, że użytkownik zgadł za 1 razem, po czym komputer zapyta czy chcemy grać dalej.
 - W przeciwnym razie komputer podpowie czy wylosowana liczba oczek jest od naszej zgadywanej mniejsza czy większa.
 - Po trafieniu powinna zostać wyświetlona zgadywana liczba oczek oraz liczba prób odgadnięcia ☺.
 - * Użytkownik powinien mieć wybór iloma kośćmi rzucamy (max 3). Wówczas odgadujemy sumę wylosowanych oczek.
- Własna inwencja programistyczna mile widziana ☺.

Każdy program powinien:

- a) „przedstawić się”,
- b) posiadać zabezpieczenia przed błędnym wprowadzaniem danych,
- c) pytać użytkownika czy chce zakończyć działanie, czy kontynuować.

ZADANIA DODATKOWE**ZADANIE**

Napisz program rozwiązujący metodą wyznaczników układ dwóch równań z dwiema niewiadomymi. Zwrócić uwagę na przypadki, kiedy układ jest sprzeczny lub nieoznaczony. Danymi pobieranymi od użytkownika jest sześć liczb reprezentujących równania $ax+by=c$, $dx+ey=f$.