

Zadania 10

1. ([zestaw10zad1bin.cpp](#)) Zaimplementować i przeanalizować następujący algorytm.

Wypisywanie binarnej reprezentacji liczby - algorytm rekurencyjny

Dane: liczba n

Wynik: binarna reprezentacja liczby n

```
procedure bin( $n$ )
  if  $n > 0$ 
    bin( $n/2$ )
  wypisz:  $n \bmod 2$ ;
```

2. ([zestaw10zad2sumacyfr.cpp](#)) Napisać funkcję wyliczającą sumę cyfr danej liczby:
- 2.1. iteracyjnie ([sumacyfr_i](#)),
 - 2.2. rekurencyjnie ([sumacyfr_r](#)).

```
#include <iostream>
using namespace std;
...
...
main()
{
    int liczba = 234567;
    cout << "Suma cyfr liczby " << liczba << " to: " << sumacyfr_i(liczba)
          << " (iteracyjnie)" << endl;
    cout << "Suma cyfr liczby " << liczba << " to: " << sumacyfr_r(liczba)
          << " (rekurencyjnie)" << endl;
}
```

3. ([zestaw10zad3wyszukiwanierekurencyjne.cpp](#)) Napisz funkcję realizującą rekurencyjny algorytm wyszukiwania binarnego dla posortowanej listy

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

// tu wpisac definicje funkcji szukajbinarnie

main()
{
    const unsigned dllisty = 20;
    int lista[dllisty];

    // Wypelnianie listy pewnym uporządkowanym ciagiem:
    double x=.3;
    lista[0]=0;
    for (int i=1; i<dllisty; i++)
    {
        x = 4*x*(1-x);
        lista[i] = lista[i-1] + 1 + 5*x;
    }
}
```

```

    }

// Wypisanie elementow listy
for (int i=0; i<dlisty; i++)
{
    cout << lista[i] << ", ";
}
cout << endl;

// definicja elementu, ktorego szukamy:
int elementszukany = 25;
// Uzycie funkcji wyszukiwania, ktora nalezy napisac:
cout << wyszukajbinarnie(lista, 0, dlisty-1, elementszukany);
}

```

4. (**zestaw10zad4minrekurencyjnie.cpp**) Zaimplementuj następujący algorytm rekurencyjny:

Wyszukiwanie minimum - wersja rekurencyjna

Dane: początek i koniec tablicy: p , k , tablica $T[p], \dots, T[k]$

Wynik: element minimalny

```

function min( $T, p, k$ )
| if  $p=k$ 
|    $\perp$  return  $T[p]$ 
|  $m = \lfloor \frac{p+k}{2} \rfloor$ 
|  $x = \min(T, p, m)$ 
|  $y = \min(T, m+1, k)$ 
| if  $x > y$ 
|    $\perp$  return  $x$ 
| else
|    $\perp$  return  $y$ 

```

5. Zaprojektuj i zaimplementuj jakiś algorytm sortowania.