

Przypomnienie

Każdy program zaczyna się od **main**

```
int main() {  
    return 0;  
}
```

Funkcje

```
int f(int x, int a) {  
    return a*x + 3;  
}
```

```
def f(x,a):  
    return a*x + 3
```

Zmienne

```
int a = 42;
```

```
a = 42
```

Typy

```
bool is_cpp_fun = false;  
char c = 'K';  
int a = 42;  
short b = 24500;  
long b = 2138000;  
float pi = 3.14f;  
double e = 2.72;  
---  
unsigned int a = 3138000;  
signed short b = -1002;  
---  
void
```

Biblioteka standardowa

```
#include <iostream>

int main() {
    int x = 2

    std::cout << x << std::endl;
    return 0;
}
```

```
print(y)
```

```
#include <iostream>

int main() {
    int x;

    std::cin >> x;
    std::cout << x << std::endl;
    return 0;
}
```

```
x = input()
```

```
#include <stdio.h>

int main() {

    int x = 13;
    printf("%d", x);

    int a;
    scanf("%d", &a);

    return 0;
}
```


Pętla FOR

```
for(`wyrażenie początkowe`; `warunek trwania`; `wyrażenie co krok`)
```

```
for(int i = 0; i < 3; i++) {  
    // code  
}
```

Zajęcia II

Cele:

- rozwijać wiedzę na temat podstaw języka
- omówić tablice oraz łańcuchy
- omówić do-while oraz switch-case

Problem

Jak zapisać ciąg DNA?

```
char dna1 = 'T';  
char dna2 = 'G';  
char dna3 = 'T';
```

```
char dna1_compliment = find_compliment(dna1);  
char dna2_compliment = find_compliment(dna2);  
char dna3_compliment = find_compliment(dna3);
```

Tablice

```
char dna[5] = {'T', 'G', 'T', 'A', 'C'};
```

```
int measurements[] = {1024, 512, 256, 128, 128, 64};
```

```
char dna[] = "ATCCGCTAC";
```

```
char end = '\\0';
```

Tablice

```
char dna[] = "ATCCGCTAC";  
char first_T = dna[?];
```

Tablice

```
char dna[] = "ATCCGCTAC";  
char first_T = dna[1];
```

std::string

```
std::string dna = "TCC";  
std::cout << dna << " is: " << dna.length() << " chars length." << std::endl;
```


Zadanie Lab I

Napisz program w języku `C`, który wczyta od użytkownika sekwencję DNA i policzy ilość nukleotydów `'T'`

```
#include <stdio.h>

int main() {
    char dna[1000];
    int count = 0;

    int correctly_read = scanf("%999s", dna);

    if( correctly_read == 1 ) {
        for (int i = 0; dna[i] != '\0'; i++) {
            if (dna[i] == 'T') {
                count++;
            }
        }
    }

    printf("%d\n", count);
    return 0;
}
```

Zadanie Lab II

Napisz program w języku `C++`, który wczyta od użytkownika sekwencję DNA i policzy ilość nukleotydów `'A'`

```
#include <iostream>
#include <string>

int main() {
    std::string dna;
    std::cin >> dna;

    int count = 0;
    for( size_t i = 0; i < dna.length(); i++ ) {
        if( dna[i] == 'A' ) {
            count++;
        }
    }

    std::cout << count << std::endl;
    return 0;
}
```

Pętla do-while

```
do {  
    // code  
} while( condition );
```

```
double pH;  
do {  
    std::cin >> pH;  
} while( pH < 0 || pH > 14 );
```

Problem

```
if( condition1 ) {  
    // do stuff 1  
} else if( condition2 ) {  
    // do stuff 2  
} else if( condition3 ) {  
    // do stuff 3  
} if ...
```

Rozwiązanie: switch-case!

```
switch( value ) {  
    case value1:  
        // do stuff 1  
        break;  
  
    case value2:  
        // do stuff 2  
        break;  
  
    default:  
        // do stuff other than previous  
        break;  
}
```

```
char nucleotide = 'A';

// print complementary nucleotide
switch( nucleotide ) {
    case 'a':
    case 'A':
        std::cout << 'T';
        break;

    ...

    default:
        std::cout << "Wprowadzono błędny nukleotyd" << std::endl;
        break;
}
```



```
char complementary(char nucleotide) {  
    switch( nucleotide ) {  
        case 't':  
        case 'T':  
            return 'A';  
  
        ...  
    }  
}
```

Zadanie I

Napisz program, który przyjmie łańcuch znaków sekwencji DNA (o maksymalnej długości 1024 znaków) i na wyjściu wydrukuje ile w danej sekwencji było nukleotydów A,T,C,G (w ten kolejności, oddzielone myślnikiem).

- Pamiętaj obsłużyć małe jak i wielkie litery.
- Pamiętaj sprawdzić, czy sekwencja jest poprawna (tzn. nie zawiera innych symboli niż oczekiwane).

Przykładowe wejście:

"ATcGAGg"

Przykładowe wyjście:

2-1-1-3

Zadanie II

Napisz program, który dokona transkrypcji podanej sekwencji DNA na ciąg RNA.

- Pamiętaj o małych i wielkich literach oraz odrzuceniu niepoprawnego ciągu wejściowego.

Przykładowe wejście:

"TGGAACTT"

Przykładowe wyjście:

"UGGAACUU"

Zadanie III

Napisz program, który przyjmie sekwencję DNA i wydrukuje ciąg komplementarny do podanego, w odwrotnej kolejności.

Przykładowe wejście:

```
"AAAACCCGGT"
```

Przykładowe wyjście:

```
"ACCGGGTTTT"
```