

Algorytmy i programowanie

Marcin Żurowski

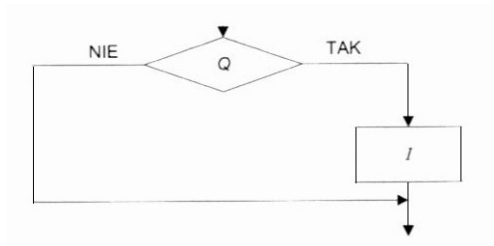
15 marzec 2021

Plan zajęć

1 Instrukcje sterujące

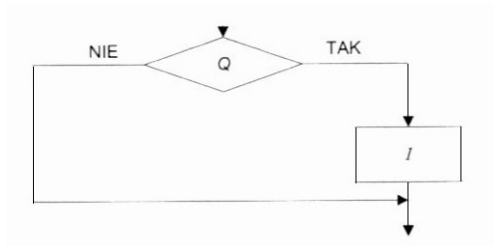
2 Zadania

Instrukcja warunkowa



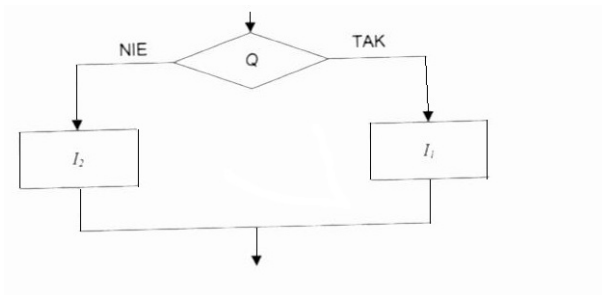
```
if (<warunek Q>){  
  <instrukcje I>  
}
```

Instrukcja warunkowa



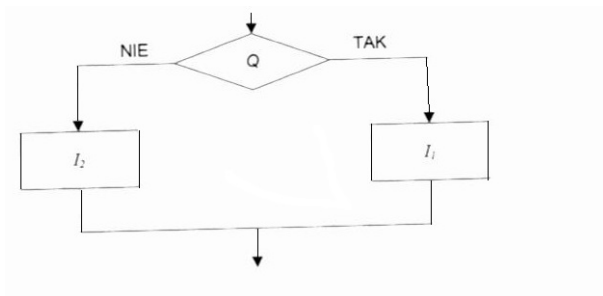
```
if (<warunek Q>){  
  <instrukcje I>  
}
```

Instrukcja warunkowa



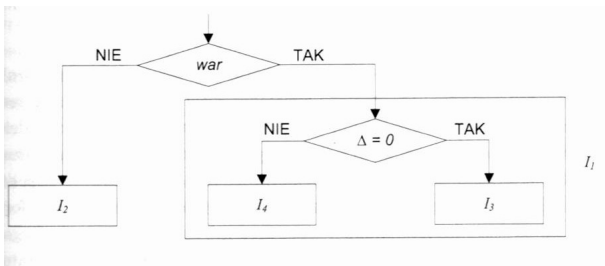
```
if (<warunek Q>){  
  <instrukcje I1>  
} else {  
  <instrukcje I2>  
}
```

Instrukcja warunkowa



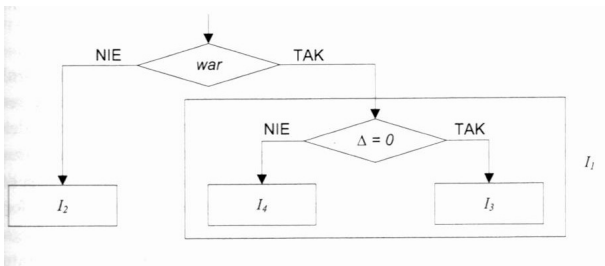
```
if (<warunek Q>){  
  <instrukcje I1>  
} else {  
  <instrukcje I2>  
}
```

Instrukcja warunkowa



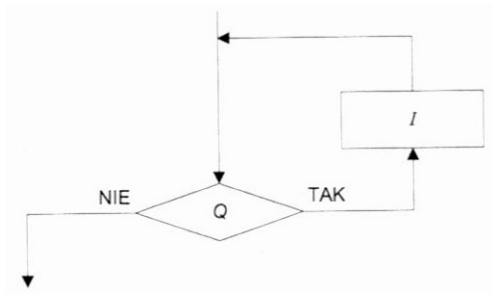
```
if (<warunek war>){  
  if (<warunek delta == 0>){  
    <instrukcje I3>  
  } else {  
    <instrukcje I4>  
  }  
} else {  
  <instrukcje I2>  
}
```

Instrukcja warunkowa



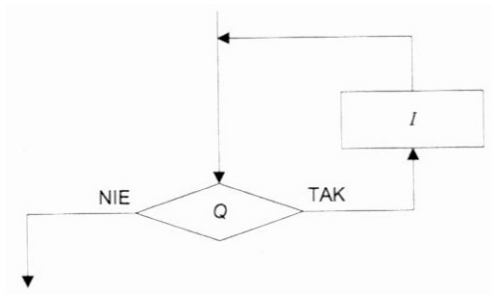
```
if (<warunek war>){  
    if (<warunek delta == 0>){  
        <instrukcje I3>  
    } else {  
        <instrukcje I4>  
    }  
} else {  
    <instrukcje I2>  
}
```


Iteracje - while



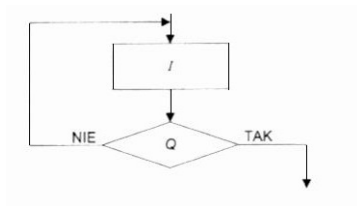
```
while (<warunek Q>){  
  <instrukcje I>  
}
```

Iteracje - while



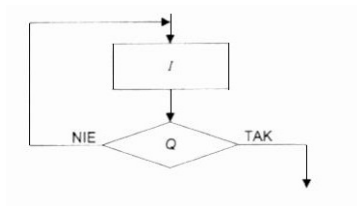
```
while (<warunek Q>){  
  <instrukcje I>  
}
```

repeat vs do



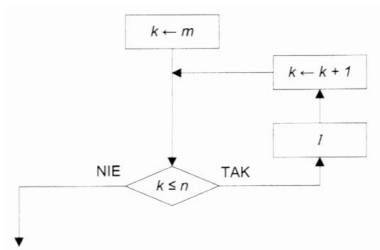
```
do {  
  <instrukcje I>  
} while (<warunek !Q>);
```

repeat vs do



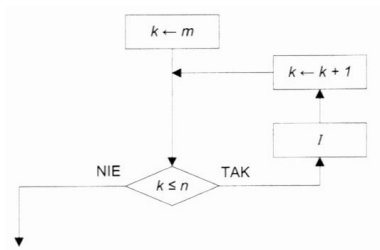
```
do {  
  <instrukcje I>  
} while (<warunek !Q>);
```

for



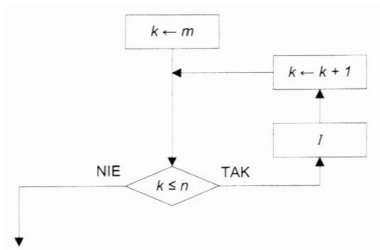
```
for (<instrukcja początkowa>; <początek
iteracji>; <koniec iteracji>){
  <instrukcje I>
}
for (int k = m; k <= n; k++){
  <instrukcje I>
}
```

for



```
for (<instrukcja początkowa>; <początek
iteracji>; <koniec iteracji>){
  <instrukcje I>
}
for (int k = m; k <= n; k++){
  <instrukcje I>
}
```

for



```
for (<instrukcja początkowa>; <początek
iteracji>; <koniec iteracji>){
  <instrukcje I>
}
for (int k = m; k <= n; k++){
  <instrukcje I>
}
```

break

```
for (int i = 0; i < 5; i++){  
    if (i == 3){  
        break;  
    }  
    cout << i << " ";  
}  
0 1 2
```


break

```
for (int i = 0; i < 5; i++){  
    if (i == 3){  
        break;  
    }  
    cout << i << " ";  
}  
0 1 2
```

continue

```
for (int i = 0; i < 5; i++){  
    if (i == 3){  
        continue;  
    }  
    cout << i << " ";  
}
```

```
0 1 2 4
```

continue

```
for (int i = 0; i < 5; i++){  
    if (i == 3){  
        continue;  
    }  
    cout << i << " ";  
}  
0 1 2 4
```

4

Wyobraźmy sobie hipotetyczną sytuację, że każdy klient w supermarkecie podchodząc do kasy, mówi, ile ma w koszyku towarów. Kasjerka z kodu kreskowego wprowadza cenę każdego produktu. Skonstruujemy algorytm, który jako wynik poda ostateczną kwotę, jaką musi zapłacić klient.

5

Podać algorytm wczytujący liczby różne od zera i obliczający ich sumę. Nie wiemy przy tym, ile liczb jest do wczytania.

6

Dane są zmienne całkowite nieujemne a oraz b . Stosując jedynie operację dodawania, napisać algorytm wyznaczający wartość wyrażenia $a * b$.

Nie używając operatora MOD, napisać w pseudokodzie algorytm badający parzystość dowolnej liczby całkowitej.

Dane są: nieujemna liczba całkowita a oraz dodatnia liczba całkowita b . Przedstawić algorytm wyznaczający iloraz q oraz resztę r z dzielenia a przez b .

Zadana jest liczba naturalna n . Podać ciąg instrukcji wypisujących dziesiętną reprezentację tej liczby w odwrotnym porządku. Na przykład dla $n = 170673$, algorytm powinien wypisać liczbę 376071.

10

Zadana jest liczba naturalna n . Podać ciąg instrukcji wypisujący dziesiętną reprezentację tej liczby od początku.

11

Dana jest liczba naturalna n . Podać algorytm wypisujący wartości kolejnych kwadratów liczb naturalnych aż do n^2 i obliczający ich sumę, to jest $1^2 + 2^2 + 3^2 + \dots + n^2$.

12

Skonstruować algorytm wypisujący liczby pierwsze, których iloczyn jest równy zadanej liczbie naturalnej $n > 0$.

Dana jest liczba naturalna $n > 1$. Skonstruować algorytm sprawdzający, czy dana liczba naturalna jest liczbą pierwszą.