13. Uzupełnienie o językach C/C++

1 Instrukcja `#define`

W języku C do utworzenia stałej używa się instrukcji preprocesora `#define`, np.:

```
#define PI 3.14159
#define LICZBA_DNI_W_TYG 7
```

Wszystkie wystąpienia ciągu znaków `PI` w kodzie programu zostaną zastąpione liczbą 3.14159 jeszcze przed właściwą kompilacją. Jednak w takiej sytuacji kompilator nie jest w stanie kontrolować poprawności powyższych definicji.

Można również tworzyć bardziej złożone definicje:

```
1  #include <iostream>
2  #define FOR(n) for (int i=0; i<n; i++)
3
4  void fun() { /* instrukcje */ }
5
6  int main() {
7    int x;
8    std::cin >> x;
9    FOR(x) fun();
10   return 0;
11 }
```

Pętla w linii 9. zostanie zamieniona przez preprocesor na:

```
for (int i=0; i<n; i++) fun();
```

W języku C++ możliwe jest zdefiniowanie stałej poprzez modyfikator `const`, np.:

```
const float PI = 3.14159;
const int LICZBA_DNI_W_TYG = 7;
```

Wartości tak zdefiniowanych zmiennych nie można zmieniać w trakcie działania programu — kompilator zgłosi to jako błąd.

2 Operator trójargumentowy `if-else`

Operator trójargumentowy `if-else` jest zwięzłym odpowiednikiem konstrukcji `if-else`, lecz w przeciwieństwie do niej zwraca wartość:

```
int zmienna;
if (warunek) {
  zmienna = wartosc_jesli_prawda;
} else {
  zmienna = wartosc_jesli_falsz
}
```

```
int zmienna = warunek ? wartosc_jesli_prawda : wartosc_jesli_falsz;
```

Przykład:

```
int maximum(int a, int b) {
  return (a > b) ? a : b;
}
```

3 Instrukcja `switch`

Instrukcja `switch` wykonuje jedną instrukcję na podstawie dopasowanej wartości (koniecznie będącą liczbą całkowitą):

```
switch(zmienna) {
  case wartosc_calkowita_1 :
    instrukcje;
    break;
  case wartosc_calkowita_2 :
    instrukcje;
    break;
  case wartosc_calkowita_3 :
    instrukcje;
    break;
  (...)
  default :
    instrukcje;
}
```

Instrukcja porównuje wartość `zmiennej` z każdą wartością całkowitą występującą po słowie kluczowym `case`. W przypadku dopasowania wykonane zostają odpowiednie instrukcje w bloku aż do najbliższego słowa kluczowego `break`. Gdyby nie było instrukcji `break`, wykonany zostałby również kod przypisany do następnego z kolei warunku. Natomiast instrukcje po słowie `default` są wykonane, jeżeli żadne z poprzednich porównań nie było prawdziwe.

Przykład:

```
//...
switch(op) {
  case '+' :
    res = a + b;
    break;
  case '-' :
    res = a - b;
    break;
  case '*' :
    res = a * b;
    break;
}
```

4 Argumenty programu

Funkcja `main` może występować w dwóch postaciach:

- `int main()`
- `int main(int argc, const char* argv[])`

Argument `argc` jest liczbą nieujemną informującą o długości tablicy `argv`. Tablica `argv` przechowuje łańcuchy znaków i zawiera listę argumentów przekazanych podczas uruchomienia programu. Wyjątkiem jest pierwszy element, którym zawsze jest nazwa uruchamianego programu.

```
for (int i=0; i<argc; i++) {
  std::cout << argv[i] << std::endl;
}
```

Aby uzyskać np. liczby całkowite z łańcuchów znaków z tablicy `argv` można wykorzystać funkcję `atoi`.

5 Liczby pseudolosowe

Do generowania liczb pseudolosowych służy funkcja `rand` (z biblioteki `stdlib.h`), która zwraca liczbę całkowitą z przedziału od 0 do `RAND_MAX`. Algorytm użyty do generowania liczb pseudolosowych wymaga zainicjalizowania poprzez funkcję `srand`, w przeciwnym wypadku przy każdym uruchomieniu programu dostaniemy takie same wartości.

Najczęściej funkcję `srand` wywołuje się z argumentem będącym liczbą sekund, które upłynęły od dnia 1 stycznia 1970 roku. Tę wartość uzyskuje się poprzez funkcję `time` z biblioteki `time.h`.

Przykład:

```
#include <iostream>
#include <cstdlib>
#include <ctime>

int main() {
  srand(time(NULL));

  std::cout << "Liczba pseudolosowa: "
    << rand() << std::endl;

  std::cout << "Liczba pseudolosowa z przedzialu 0..99: "
    << rand() % 100 << std::endl;
  std::cout << "Liczba zmiennoprzecinkowa: "
    << (float)rand() / (float)RAND_MAX << std::endl;

  return 0;
}
```

6 Zadanie — zgadnij liczbę! (0-3 punkty)

Napisz program, który umożliwi grę w zgadywanie wylosowanej liczby. Program losuje liczbę x z zakresu od a do b, a użytkownik zgaduje jaka to liczba. Jeżeli podana przez gracza liczba będzie większa od x, program wypisze napis "mniej!", jeżeli mniejsza — napis "wiecej!". Gra się kończy napisem "trafiono!", gdy liczba x zostanie odgadnięta.

Liczby a i b są podawane przez użytkownika jako argumenty podczas uruchamiania programu. Możliwe jest również podanie tylko jednej wartości, wówczas liczbę losujemy z przedziału $0..a$.

Uruchomienie programu z argumentem postaci `-h` lub `-help`, powoduje wypisanie prostej pomocy programu wraz z imieniem, nazwiskiem i numerem indeksu autora, oraz jego zakończenie bez uruchamiania gry.

Przykładowe uruchomienia programu:

```
./zgadnijliczbe.exe 3 15      // zgadywanie liczby z przedzialu 3..15
./zgadnijliczbe.exe 5         // zgadywanie liczby z przedzialu 0..5
./zgadnijliczbe.exe --help    // tylko wypisanie autora programu
./zgadnijliczbe.exe 3 12 -h   // tylko wypisanie autora programu
./zgadnijliczbe.exe -h 3 12   // tylko wypisanie autora programu
```