

Ustalenia techniczne XXVI OI

Do kompilowania rozwiązań w języku C++ używany będzie następujący kompilator:

- G++ 4.8

Polecenie używane do kompilacji rozwiązań w języku C++ (np. zadania abc):

- `g++ -O3 -static abc.cpp -std=c++11`

Do uruchamiania rozwiązań w języku Python używany będzie interpreter:

- Python 3.4

Polecenie używane do uruchomienia rozwiązania w języku Python (zadanie abc):

- `python3 abc.py`

W przypadku języka Python, komunikat „błąd kompilacji” oznacza błąd składni.

Nie jest możliwy inny sposób kompilacji (np. przy pomocy plików Makefile). Programy, które nie będą kompilować się w podany sposób, **nie podlegają ocenie**.

Ograniczenia:

- kod źródłowy rozwiązania nie może przekraczać 100 KB, a kod wykonywalny 10 MB,
- czas kompilacji rozwiązania nie może przekraczać 30 s,
- wielkość pamięci operacyjnej dostępnej programom będzie podana w treści zadań (jednak ta wartość dotyczy sumarycznego zapotrzebowania na pamięć, a więc zawiera m.in. rozmiar kodu wykonywalnego, stosu, sterty itp.),
- maksymalny i minimalny limit czasowy z testów dla każdego podzadania będzie podany w systemie SIO.

Rozwiązania powinny:

- składać się z jednego pliku źródłowego o nazwie podanej w treści zadania,
- czytać dane ze standardowego wejścia, zapisywać wynik na standardowe wyjście, chyba że dla danego zadania wyraźnie napisano inaczej,
- kończyć działanie kodem wyjścia 0 – każdy inny kod zakończenia programu jest równoważny błędowi wykonania,
- zakładać, że są wykonywane na procesorze 32-bitowym (w szczególności, że typ `int` w C++ jest 32-bitowy).

Rozwiązania mogą:

- korzystać z biblioteki standardowej,
- być napisane w standardzie C++11,
- korzystać z biblioteki NumPy w języku Python,
- wypisywać na standardowe wyjście diagnostyczne `stderr` (komunikaty wypisane w ten

sposób są ignorowane przez system sprawdzający; należy jednak pamiętać, że takie wypisywanie zużywa cenny czas).

Rozwiązania nie mogą:

- otwierać jakichkolwiek plików (w szczególności zabronione jest tworzenie plików tymczasowych),
- tworzyć nowych procesów czy wątków,
- korzystać z funkcji sieciowych (np. `socket`, `send` itp.),
- naruszać bezpieczeństwa systemowego,
- oczekiwać na interakcję użytkownika,
- uruchamiać innych programów,
- korzystać z zewnętrznych bibliotek (oprócz dozwolonych bibliotek dla C++ i Pythona), chyba że treść zadania wskazuje inaczej. Dotyczy to również bibliotek dołączanych przez opcje kompilatora.

Rozwiązania w języku C++:

- nie powinny używać plików nagłówkowych typowych dla środowiska DOS/Windows, np. `conio.h` czy `windows.h`,
- nie powinny używać niestandardowych argumentów do funkcji `printf`, takich jak `%D` czy `%F`,
- w przypadku korzystania tylko z wejścia/wyjścia strumieniowego (`iostream`) radzimy na początku programu wykonywać instrukcje:
`std::ios_base::sync_with_stdio(false);`
`std::cin.tie(NULL);`
oraz wypisywać końce linii jako `"\n"` (a nie `endl`). Służy to zwiększeniu efektywności operacji wejścia/wyjścia.