Zadanie: ZGA
Zgadywanka

Etap III, dzień próbny, plik źródłowy `zga.*`, dostępna pamięć 32 MB *30-03-2004*

Bajtazar stał się nałogowym hazardzistą. Wszystkie pieniądze przepuszcza w kasynie grając w „zgadywankę". Twierdzi, że można opracować system gry pozwalający wygrać z kasynem.

Jedna rozgrywka zgadywanki polega na tym, że losujemy kolejno 9 różnych liczb rzeczywistych z przedziału $(0;1)$, przy jednostajnym rozkładzie prawdopodobieństwa. Bezpośrednio po wylosowaniu każdej liczby należy od razu określić, która co do wielkości jest to liczba. Oczywiście nie da się tego przewidzieć, trzeba próbować to zgadnąć. Jeżeli trafnie określi się uporządkowanie wszystkich 9 liczb, wygrywa się. W przeciwnym przypadku przegrywa się. Jeśli, na przykład, pierwszą wylosowaną liczbę określimy jako drugą co do wielkości, a potem wylosujemy jeszcze dwie liczby mniejsze od niej, to przegramy.

Zadanie

Pomóż Bajtazarowi! Zaprogramuj jak najlepszą strategię grania w zgadywankę. Napisz program, który rozegra wiele rozgrywek opisanej gry i wygra jak najwięcej razy. Ocena Twojego programu będzie tym wyższa, im więcej rozgrywek on wygra. Dokładniej, za każdą wygraną kasyno płaci graczowi 423,99 bajtalarów, natomiast za każdą przegraną pobiera od gracza 13,53 bajtalarów. Twój program rozegra 10^6 rozgrywek. Otrzymasz tyle punktów, ile wynosi całkowity uzyskany zysk podzielony przez 10^4 i zaokrąglony do najbliższej liczby całkowitej z przedziału $[0, 100]$.

Musisz zaprogramować moduł zawierający następujące trzy procedury i funkcje:

- `procedure inicjalizuj` / `void inicjalizuj()` – ta procedura zostanie wywołana tylko raz, na początku, przed rozegraniem wszystkich rozgrywek; możesz jej użyć do zainicjalizowania swoich struktur danych.
- `procedure nowa_rozgrywka` / `void nowa_rozgrywka ()` – ta procedura będzie wywołana na początku każdej rozgrywki. Możesz jej użyć by zainicjalizować zmienne związane z jedną rozgrywką.
- `function kolejna_liczba (x : Double) : Integer` / `int kolejna_liczba (double x)` – ta funkcja dostaje jako parametr kolejną wylosowaną liczbę x; $0 < x < 1$, podaną z precyzją do 12 miejsc dziesiętnych. Funkcja powinna obliczyć, która co do wielkości jest to liczba wśród 9 liczb losowanych w aktualnej rozgrywce i zwrócić wynik. Jeśli rozgrywka jest przegrana, funkcja może zwrócić dowolną liczbę całkowitą od 1 do 9.

Pliki (Pascal)

W katalogu `zga_pas` znajdziesz następujące pliki:

- `zga.pas` – szkielet modułu grającego zawierający puste procedury i funkcje `inicjalizuj`, `nowa_rozgrywka`, `kolejna_liczba`. Powinieneś napisać kod tych funkcji.

- `kasyno.pas` – program generujący wiele rozgrywek (tyle ile określono w stałej `ILE_ZESTAWOW`) zgadywanki. Każda rozgrywka jest rozgrywana z użyciem procedur i funkcji modułu grającego `zga.pas`. Możesz użyć tego programu do testowania swojego modułu grającego.
- `makefile` – plik umożliwia skompilowanie programu `kasyno.pas` z dołączonym modułem `zga.pas` za pomocą polecenia `make`.

Pliki (C/C++)

W katalogu `zga_c` / `zga_cpp` znajdziesz następujące pliki:

- `zga.h` – plik zawierający nagłówki funkcji `inicjalizuj`, `nowa_rozgrywka` i `kolejna_liczba`.
- `zga.c` / `zga.cpp` – szkielet modułu grającego zawierający puste definicje funkcji zadeklarowanych w pliku nagłówkowym `zga.h`. Powinieneś napisać kod tych funkcji.
- `kasyno.c` / `kasyno.cpp` – program generujący wiele rozgrywek (tyle ile określono w stałej `ILE_ZESTAWOW`) zgadywanki. Każda rozgrywka jest rozgrywana z użyciem procedur i funkcji modułu grającego `zga.c` / `zga.cpp`. Możesz użyć tego programu do testowania swojego modułu grającego.
- `makefile` – plik umożliwia skompilowanie programu `kasyno.c` / `kasyno.cpp` z dołączonym modułem `zga.c` / `zga.cpp` za pomocą polecenia `make`.

Rozwiązanie

Wynikiem Twojej pracy powinien być tylko jeden plik `zga.c` albo `zga.cpp` albo `zga.pas`.

Przykład

Przykładowa interakcja z programem może wyglądać następująco (pochyłym krojem zaznaczono liczby zwracane przez funkcję `kolejna_liczba`; rozegrano 2 rozgrywki):

Rozgrywka 1

0.043319026120
1
0.933801434041
8
0.992050359372
9
0.145189967093
4
0.518803250649
6
0.093583048537
2
0.764309529654

7
0.338653748790
5
0.119437652934
3 Wygrana

Rozgrywka 2

0.164020610610
2
0.205594263575
3
0.042637391231
1
0.147521628974
1 Przegrana
0.946549875333
1
0.772946216573
1
0.152956276544
1
0.539653928563
1
0.552047936535
1