

SPIS TREŚCI

NUMERU 9 (153)

Ewolucja w sposobie korzystania z komputerów w ostatnim ćwierćwieczu <i>Doc. dr Jan Madey</i>	str. 1
Patrz w niebo	str. 3
Powiększenia mikrokomputerowe <i>Dr hab. Jerzy Jurkiewicz</i>	str. 4
Czego nie potrafi komputer? <i>Mgr Jarosław Deminet</i>	str. 5
Myszka i okienko <i>Mgr Jarosław Deminet</i>	str. 6
Mała Delta	str. 8
Metoda Monte Carlo a ruchy Browna <i>Dr hab. Ryszard Kutner</i>	str. 10
Zadania	str. 12
Rysowanie prostej nie jest proste <i>Dr Michał Jankowski</i>	str. 13
Małe bazy danych w mikrokomputerze <i>Mgr Jan Raszewski</i>	str. 14
Klub 44	str. 16

W następnym numerze:

O protonie

„Delta”

matematyczno-fizyczno-astronomiczny
miesięcznik popularny
Polskiego Towarzystwa
Matematycznego, Polskiego
Towarzystwa Fizycznego i Polskiego
Towarzystwa Astronomicznego
wydawany przy poparciu
Ministerstwa Oświaty i Wychowania

Komitet Redakcyjny

dr Maciej Bryński
dr Bogdan Cichocki
dr Alicja Derkowska
doc. dr Jan A. Gaj
doc. dr Bolesław Gleichgewicht
doc. dr Tomasz Hofmoki
doc. dr Tadeusz Jarzembowski
doc. dr Marcin Kubiak
mgr Andrzej Mąkowski
dr Zbigniew Płochocki — v-przewodniczący
dr Jan Rempała
prof. dr Konrad Rudnicki
prof. dr Grzegorz Sitarski
prof. dr Józef I. Smak
prof. dr Kazimierz Stepien
prof. dr Mieczysław Subotowicz
dr Michał Szurek
doc. dr Andrzej Szymacha
doc. dr Aniela Wołska
prof. dr Andrzej Woszczyk
prof. dr Wojciech Żakowski —
przewodniczący

WARUNKI PRENUMERATY

Cena prenumeraty kwartalnej zł 105,— półrocznej zł 210,— rocznej zł 420,—

- dla osób prawnych — instytucji i zakładów pracy:
 - instytucje i zakłady pracy zlokalizowane w miastach wojewódzkich i pozostałych miastach, w których znajdują się siedziby oddziałów RSW „Prasa-Książka-Ruch”, zamawiają prenumeratę w tych oddziałach,
 - instytucje i zakłady pracy zlokalizowane w miejscowościach, gdzie nie ma oddziałów RSW „Prasa-Książka-Ruch” i na terenach wiejskich opłacają prenumeratę w urzędach pocztowych i u doręczycieli.
- dla osób fizycznych — indywidualnych prenumeratorów:
 - osoby fizyczne zamieszkałe na wsi i w miejscowościach, gdzie nie ma oddziałów RSW „Prasa-Książka-Ruch”, opłacają prenumeratę w urzędach pocztowych i u doręczycieli,
 - osoby fizyczne zamieszkałe w miastach — siedzibach oddziałów RSW „Prasa-Książka-Ruch” opłacają prenumeratę wyłącznie w urzędach pocztowych nadawczo-oddawczych właściwych dla miejsca zamieszkania prenumeratora. Wpłaty dokonują używając „blankietu wpłaty” na rachunek bankowy miejscowego oddziału RSW „Prasa-Książka-Ruch”.
- Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW „Prasa-Książka-Ruch”, Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto NBP XV Oddział w W-wie Nr 1153-201045-139-11. Prenumerata ze zleceniem wysyłki za granicę pocztą zwykłą jest droższa od prenumeraty krajowej o 50% dla zlecających instytucji i zakładów pracy. Terminy przyjmowania prenumeraty na kraj i za granicę:
 - do dnia 10 listopada na I kwartał, I półrocze roku następnego oraz cały rok następny,
 - do dnia 1-go każdego miesiąca poprzedzającego okres prenumeraty roku bieżącego.

Cena 1 egzemplarza zł 35,—

Redaguje kolegium w składzie:

mgr inż. Krzysztof Biesaga
mgr Maciej Jędrzejczak — z-ca red. nac.
mgr Krystyna Kordos — sekr. red.
dr hab. Marek Kordos — red. nac.
dr Tomasz Kwast — z-ca red. nac.
mgr inż. arch. MariąŁa Łotysz
dr Andrzej Majhofer
mgr Anna Rudnik
dr Jerzy Ryll
mgr Joanna Udalska
mgr Jan Zalewski

Adres Redakcji
ul. Koszykowa 6a
00-564 Warszawa
tel. 21-19-85

Krajowe Wydawnictwo Czasopism

RSW „Prasa—Książka—Ruch”
ul. Noakowskiego 14
00-666 Warszawa
tel. centr. 25-72-91 do 93
Biuro Reklam i Propagandy
tel. 25-56-26
Nakład 25 000 egz. Objętość 2 ark. wyd.;
2,50 ark. druk;
papier offsetowy V kl. 70 g.
Wydrukowano w Drukarni
im. Rewolucji Październikowej
Warszawa, ul. Mińska 65.
Nr zam. 2541/12/86. P-53.

Ewolucja w sposobie korzystania z komputerów w ostatnim ćwierćwieczu

Doc. dr Jan MADEY

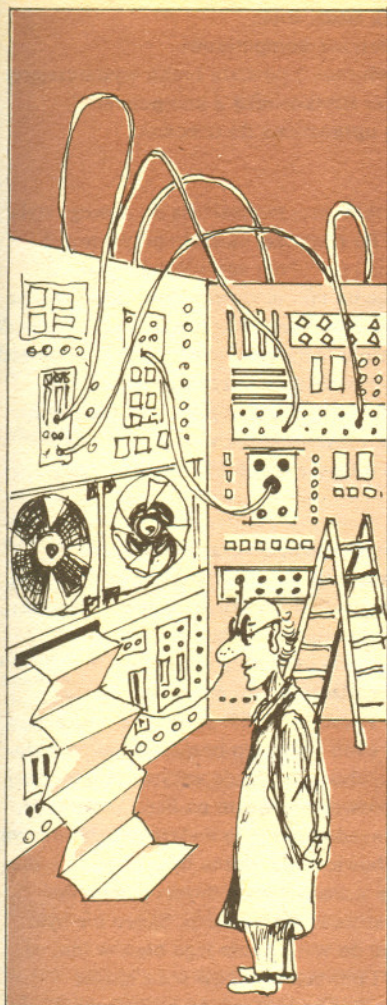
Pierwsze komputery budziły w pełni zasłużony szacunek. Duże, hałaśliwe, migające, tajemnicze, niezrozumiałe. Były one dostępne tylko dla wąskiego grona specjalistów, których praca była także tajemnicza i niezrozumiała. Ale to już prehistoria informatyki — komputery z tego okresu są cennymi eksponatami muzealnymi.

Znaczące upowszechnienie komputerów nastąpiło na przełomie lat pięćdziesiątych i sześćdziesiątych i było wynikiem skoku technologicznego (zarówno w elektronice, jak i w informatyce), który miał miejsce w poprzedzających latach. Pojawiły się mianowicie komputery nowej generacji, konstruowane z zastosowaniem tranzystorów i rdzeni ferrytowych. Dzięki temu zwiększyła się znacznie ich niezawodność, szybkość działania, pojemność pamięci, a przy tym zmniejszyły się ich gabaryty oraz — co bardzo istotne — cena. Jednocześnie powstały pierwsze języki programowania wysokiego poziomu, co pozwoliło na znaczne poszerzenie kręgu osób zainteresowanych programowaniem lub po prostu użytkowaniem komputerów.

Przykładem komputera tamtych dni jest (a raczej był) GIER produkcji duńskiej firmy Regnecentralen, który przez wiele lat stanowił jedyne wyposażenie ośrodka obliczeniowego Uniwersytetu Warszawskiego. Zakupiony w końcu 1963 roku pracował praktycznie „na okrągło” przez ponad 15 lat, wspomagając początkowo cały Uniwersytet Warszawski, a później już tylko Instytut Informatyki UW. Prześledźmy na tym typowym przykładzie ewolucję w sposobie korzystania z komputerów, która odbyła się w okresie ćwierćwiecza.

Parametry GIERa budzą obecnie zdumienie — był on przecież w chwili zakupu szczytem osiągnięć technologicznych w klasie średnich komputerów. Pamięć operacyjna miała pojemność 1024 słowa (42 bitowe)! Przeliczając to na popularne obecnie bajty otrzymujemy zaledwie 5 kB (dla porównania, najpopularniejszy teraz w Polsce mikrokomputer Spectrum w najtańszej konfiguracji ma 16 kB pamięci). Do tego dochodził bęben magnetyczny o pojemności 12800 słów i ... przez kilka pierwszych lat to było wszystko. Okazuje się, że pamięć taka wystarczyła nie tylko firmie na opracowanie kompilatora języka Algol 60 o reputacji najlepszego na świecie, ale także liczny użytkownikom GIERa na prowadzenie różnego rodzaju zaawansowanych obliczeń. Kontakt z komputerem odbywał się na dwa sposoby: albo poprzez konsolę operatorską, albo za pośrednictwem taśmy papierowej. W pierwszym przypadku mamy analogię do popularnych obecnie monitorów z klawiaturą. Tyle tylko, że nie było ekranu — konsola to po prostu elektryczna maszyna do pisania. Drugi rodzaj kontaktu z komputerem miał na celu przyspieszenie operacji wprowadzania informacji i wyprowadzania wyników. Taśmę z tekstem programu lub z danymi do programu perforowało się na specjalnym urządzeniu zwanym Flexowriter, które nie było podłączone bezpośrednio do komputera. Była to znowu elektryczna maszyna do pisania, tyle że wyposażona dodatkowo w perforator oraz w czytnik taśmy papierowej. Wcześniej przygotowaną taśmę wczytywało się do komputera poprzez bardzo szybki czytnik (2000 znaków na sekundę, czyli około 5 metrów taśmy na sekundę). Wyniki programu otrzymywało się (przy trybie pracy z omijaniem konsoli) również w postaci taśmy, która była perforowana z szybkością do 150 znaków na sekundę przez podłączony do GIERa perforator i którą następnie trzeba było odczytywać na Flexowriterze. Po kilku latach dorobiliśmy się wygodniejszego urządzenia wyjściowego — drukarki wierszowej. Przyspieszyło to około 10 razy drukowanie wyników.

W pierwszych latach dostęp do GIERa uzyskiwało się rezerwując po prostu dla siebie całą maszynę na określony czas. Co się wówczas robiło, było słodką tajemnicą danej osoby. Tak poważniej, to w czasie takiej sesji zarówno uruchamiano się programy, jak i je eksploatowało. Pódbobnie jak dzisiaj na mikrokomputerach. Ale pamiętajmy, że były to lata sześćdziesiąte i cena komputera klasy GIERa wraz z urządzeniami pomocniczymi wynosiła sumę rzędu 200 tysięcy dolarów, a przy tym dolar miał znacznie wyższą niż obecnie wartość. Czyli, innymi słowy, koszt sesji przy komputerze był bardzo wysoki, a przy tym chętnych było dużo, a sprzętu mało. Nie była to wyłącznie specyfika Uniwersytetu Warszawskiego; podobna sytuacja istniała praktycznie na całym świecie.



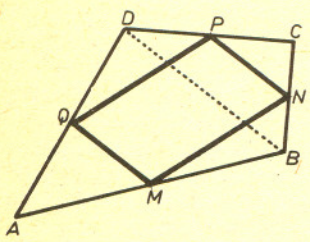


Rozwiązanie zadania M 446. Środki boków czworokąta wypukłego są wierzchołkami równoległoboku, którego pole jest równe połowie pola czworokąta, ponieważ

$$P_{PNC} + P_{AHQ} = \frac{1}{4} P_{ABCD},$$

$$P_{H\dot{A}N} + P_{QPD} = \frac{1}{4} P_{ABCD}.$$

Zatem czworokąty o pokrywających się środkach boków mają równe pola.



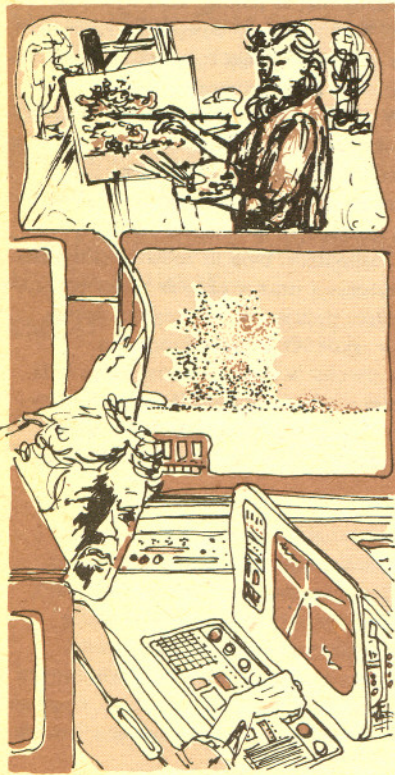
Stało się więc jasne, że taki wygodny tryb pracy — człowiek siedzi sobie przy konsoli i może się swobodnie namyślać przed każdą daną komputerowi dyspozycją — nie zdaje egzaminu z przyczyn ekonomicznych oraz z powodu braku wystarczającej liczby komputerów. Nastąpił zmierzch „intymnego kontaktu użytkownika z komputerem”, pojawiła się funkcja operatora, który stał się jedyną osobą uprawnioną do obecności w pomieszczeniu z komputerem.

Co więc musiał robić normalny użytkownik? Przede wszystkim przyzwyczać się do nowego stylu pracy, całkiem odmiennego i znacznie mniej przyjemnego. Po pierwsze, należało przygotować bardzo starannie swój program, wyperfektować go na taśmie i podobnie uczynić z danymi. Następnie trzeba było wypełnić specjalny formularz podając na nim w szczególności drobiazgowo instrukcje dla operatora — co i kiedy ma zrobić, jak ma reagować na konkretne sytuacje. Należało ponadto oszacować czas realizacji programu; po przekroczeniu tego czasu operator miał obowiązek zatrzymać komputer bez względu na to, czy zostały już wyprowadzone wszystkie wyniki. Tak przygotowany zestaw materiałów zostawiało się w ustalonym miejscu, a za jakiś czas (na ogół za długi zdaniem każdego użytkownika) otrzymywało się w innym miejscu wyniki. Taki tryb pracy określa się mianem trybu *wsadowego*, w odróżnieniu od wcześniej omawianego trybu *konwersacyjnego*.

Tryb *wsadowy* odsunął użytkownika od komputera, znacznie przedłużył czas obrotu zadania (tzn. czas, który mija od chwili oddania zadania do przetwarzania do chwili uzyskania wyników), uniemożliwił *konwersacyjną* realizację programów, istotnie utrudnił ich uruchamianie. Zwłaszcza dla osób, które przyzwyczyły się już do bezpośredniego kontaktu z komputerem, były to niezwykle niekorzystne zmiany. Ale takie były realia tamtych lat i pojawiły się modele komputerów z tzw. systemami operacyjnymi (programami nadzorującymi pracę komputera) specjalnie przystosowanymi do pracy w trybie *wsadowym*. Pocziwi GIER specjalnie „nie przejął się” tą sytuacją; może nawet „odetchnął z ulgą”, że mniej osób będzie go bezpośrednio męczyło. Czy tryb *wsadowy* miał tylko wady? Jak już wspominałem, chodziło przede wszystkim o bardziej ekonomiczne wykorzystanie zasobów komputerowych, o pozwolenie innym na liczenie w czasie, gdy się myśli. Okazało się jednak, że wprowadzenie trybu *wsadowego* było dodatkowo naturalnym bodźcem dla rozwijania metodyki programowania. Skoro bowiem nie można było łatwo i szybko sprawdzać swoich pomysłów na komputerze, to należało je wcześniej gruntowniej przemyśleć i dopiero takie dopracowane już programy zostawiać operatorowi do realizacji. Skoro nie było szansy na częste testowanie programu, to trzeba było dążyć do wyszukania jego błędów innymi metodami. Lata siedemdziesiąte przyniosły wiele ważnych i nowych wyników w metodyce programowania i sądzę, że zawdzięczamy to w dużym stopniu owemu trudnemu okresowi, kiedy dominowało przetwarzanie *wsadowe*.

Kolejna zmiana to masowe pojawienie się na przełomie lat sześćdziesiątych i siedemdziesiątych komputerów z wieloprogramowością. Komputery takie mogły w jednej chwili obsługiwać wiele osób. Zniknęły więc główne powody wprowadzenia *wsadowego* trybu przetwarzania; gdy dana osoba myślała siedząc przy jednej z wielu konsoli, komputer realizował w tym czasie inny program. Takie możliwości miał na przykład następca GIERa, system RC 4000, który choć zawiązał do Polski, to, niestety, nie na Uniwersytet Warszawski. Niezależnie od wieloprogramowości komputery tej nowej generacji miały znowu wiele zmian. W zakresie sprzętu znacznie poprawiły się parametry szybkościowe i pojemnościowe, wzbogacone zostały urządzenia zewnętrzne, znowu zmniejszyły się gabaryty, ale pojawiły się bardzo rozbudowane i w rezultacie kosztowne instalacje. Jeżeli chodzi o narzędzia programistyczne, to przede wszystkim trzeba odnotować rozwój języków programowania, a w szczególności powstanie języka Pascal, który z czasem stał się niekwestionowanym światowym standardem, zwłaszcza w zakresie nauczania programowania. Jak więc wygląda dostęp do komputera? Stosuje się zarówno tryb *wsadowy*, jak i *konwersacyjny*, i to czasami nawet w przypadku tego samego komputera. Tryb *konwersacyjny* różni się jednak od tego z „dawnych dobrych czasów”. Użytkownik siedzi wprawdzie przy konsoli lub monitorze ekranowym, ale nie jest to w sąsiedztwie komputera; wręcz przeciwnie, nawet często w znacznym oddaleniu od niego. Nie sprzyja to ponownemu nawiązaniu „intymnego kontaktu” i nie zachęca do głębszego zrozumienia zasad działania sprzętu. Dla jednych to lepiej, dla innych gorzej.

No i wreszcie lata obecne — rewolucja mikrokomputerowa. Sprzęt dosłownie trafia pod „strzechy”, komputery pojawiają się w domach, w rękach na ogół laików. Cechuje je łatwość obsługi, bogactwo zastosowań oraz różnorodność wygodnych i prostych narzędzi programistycznych. Typowy mikrokomputer domowy ma obecnie znacznie lepsze parametry techniczne niż GIER, a ponadto lepsze, wygodniejsze i bogatsze oprogramowanie. Nastąpił pełny nawrót do bezpośredniego obcowania użytkownika z komputerem, co sprzyja w szczególności jego odczarowaniu. Nowe pokolenie będzie traktowało komputer jako przedmiot codziennego użytku, jako składnik obecnej cywilizacji. A dla wielu dzieci w krajach rozwiniętych komputer już teraz jest urządzeniem równie naturalnym, jak telewizor. Oby ten stan zapanował możliwie szybko i w Polsce.



Mikrokomputer dnia dzisiejszego przewyższa znacznie swoich „ojców i dziadków” nie tylko pod względem ceny i gabarytu. Pojawiły się możliwości, o których dawniej się nie marzyło. Kolor i grafika jest rzeczą naturalną nawet w przypadku najtańszego sprzętu. Lepsze mikrokomputery mają ponadto nowe możliwości kontaktu z człowiekiem. Obok klawiatury pojawiają się „myszki”, czyli urządzenia ułatwiające szybkie przemieszczanie się po ekranie, oraz „manetki” (ang. joystick) wykorzystywane przede wszystkim w grach komputerowych. Dźwięk jest również typowym atrybutem mikrokomputera (choć trzeba przyznać, że nawet na GIERze można było wygrywać różne melodyjki). Pojemne i stosunkowo tanie pamięci dyskowe nowego rodzaju wyeliminowały całkowicie potrzebę stosowania takich nośników, jak taśma papierowa czy też karty perforowane. Do tego trzeba dodać, że mikrokomputer staje się częścią integralną wielu urządzeń i tym samym człowiek często nieświadomie z niego korzysta. A jaka będzie przyszłość? Przy obecnym tempie rozwoju technologii nie sposób prognozować nawet na najbliższe kilka lat. Jedno jest tylko pewne — czy to się nam podoba, czy nie, komputer wkroczył nieodwracalnie w nasze życie.

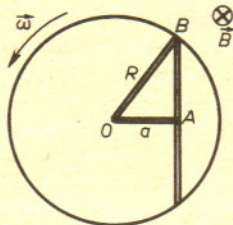


Rozwiązanie zadania F 205. Na poruszające się wraz z przewodnikiem nośniki prądu działa siła Lorentza. Powoduje ona przemieszczanie się nośników, co prowadzi do powstania różnicy potencjałów.

Wprowadzmy dwa dodatkowe przewodniki skierowane wzdłuż promienia tarczy (rysunek). Siła elektromotoryczna działająca w obwodzie OAB jest równa zero, ponieważ całkowity strumień indukcji pola magnetycznego przez ten obwód jest stały. W obwodzie nie płynie prąd. Oznacza to, że

$$(1) \quad \mathcal{E}_{AB} - \mathcal{E}_{OB} + \mathcal{E}_{OA} = 0,$$

gdzie \mathcal{E} oznacza siłę elektromotoryczną w odpowiednim fragmencie obwodu. Siła elektromotoryczna we fragmencie OB jest



$$\text{równa } \mathcal{E}_{OB} = \frac{\Delta S \cdot B}{\Delta t}, \text{ gdzie } \Delta S = \frac{\omega R^2}{2} \Delta t$$

jest polem powierzchni wycinka kołowego, który zakreślił przewodnik w czasie Δt .

Wynika stąd, że

$$\mathcal{E}_{OB} = B \frac{\omega R^2}{2}$$

i podobnie w przewodniku OA

$$\mathcal{E}_{OA} = B \frac{\omega a^2}{2},$$

gdzie $a = |OA| = \sqrt{R^2 - l^2/4}$.

Po podstawieniu do warunku (1) otrzymujemy

$$\mathcal{E}_{AB} = \mathcal{E}_{OB} - \mathcal{E}_{OA} = \frac{B\omega}{2} (R^2 - a^2) = \frac{B\omega l^2}{8}.$$

Patrz w niebo

W czasach, gdy nie znana była przyczyna zaćmień Słońca i Księżyca, zjawiska te wywoływały przerażenie i popłoch wśród ludzi. Już w starożytności umiano jednak właściwie interpretować zaćmienia, a także przewidywać czasy ich wystąpienia

W szczególności zauważono, że występują one wówczas, gdy Księżyc podczas nowiu (zaćmienie Słońca) lub pełni (zaćmienie Księżyca) znajduje się w pobliżu któregoś z węzłów swej orbity (tzn. w punkcie przecięcia orbity Księżyca z ekliptyką). W dawnych czasach wierzono, że w węźle znajduje się smok pożerający Słońce, stąd do dziś miesiącem smoczym zwiemy odstęp czasu między dwoma kolejnymi przejściami Księżyca przez ten sam węzeł.

Przy każdym przejściu Słońca przez węzeł orbity Księżyca musi wystąpić co najmniej jedno zaćmienie Słońca, a mogą zdarzyć się dwa. Maksymalnie w ciągu roku może wystąpić pięć zaćmień słonecznych, jeśli w tym czasie Słońce trzy razy przejdzie przez węzeł orbity Księżyca. Mogą zdarzyć się lata, w czasie których w ogóle nie wystąpią zaćmienia Księżyca, największa zaś ich liczba wynosić może trzy.

Choć zaćmienia Słońca występują częściej niż Księżyca, na danym obszarze Ziemi są znacznie rzadziej obserwowane. Wąski pas zaćmienia Słońca obejmuje bardzo niewielki fragment powierzchni Ziemi, a zaćmienie Księżyca może być obserwowane wszędzie tam, gdzie podczas zjawiska znajduje się on nad horyzontem.

W bieżącym roku występują dwa całkowite zaćmienia Księżyca. Pierwsze z nich miało miejsce 24 kwietnia, jednak z terenu Polski nie było widoczne. 17 października mamy szansę obejrzeć tego typu zjawisko, a warto, gdyż następne będzie w całości widoczne w Polsce dopiero 9 lutego 1990 roku.

Tegoroczne zaćmienie będzie miało następujący przebieg:

O godzinie 17^h20^m Księżyc zacznie wchodzić w półcień Ziemi, lecz istotne pociemnienie i zmianę zabarwienia tarczy będzie można dostrzec dopiero, gdy w całości zostanie ona ogarnięta przez

półcień. Pierwszy kontakt, tj. moment zewnętrznego zetknięcia tarczy ze stożkiem cienia nastąpi o godzinie 18^h29^m. Na wschodnim brzegu tarczy naszego satelity powstanie ciemne „wygryzienie”, które z czasem będzie się powiększać aż do ogarnięcia całej tarczy. Moment (19^h41^m), w którym Księżyc zetknie się wewnątrz ze stożkiem cienia (drugi kontakt), będzie początkiem zaćmienia całkowitego. W sytuacjach wyjątkowych, gdy zaćmienie jest centralne, tj. gdy Księżyc przesuwa się wzdłuż średnicy cienia, faza ta może trwać 1^h40^m. Zaćmienie tegoroczne jest mniej głębokie — Księżyc pozostanie w całości pograżony w cieniu Ziemi przez 1^h14^m, a maksimum zaćmienia, tj. najgłębsze zanurzenie nastąpi o godzinie 20^h18^m.

W czasie zaćmienia całkowitego tarcza Księżyca nie będzie niewidoczna, lecz przyjmie ciemną, czerwono-ochrową barwę. Atmosfera ziemska silnie rozprasza promieniowanie krótkofalowe, a więc w czasie trwania tej fazy do powierzchni Księżyca dociera głównie ugięte w atmosferze promieniowanie długofalowe, nadając mu charakterystyczne zabarwienie. Pociemnienie tarczy w tej fazie uzależnione jest od różnych czynników, jak np. głębokość zaćmienia, stopień zachmurzenia atmosfery. Zdarzają się zaćmienia, w czasie których tarczy Księżyca podczas centralnej fazy całkowitego jego zaćmienia w ogóle nie widać.

Całkowite zaćmienie zakończy się w chwili drugiej, wewnętrznej styczności tarczy Księżyca ze stożkiem cienia (trzeci kontakt) o godzinie 20^h55^m. Księżyc zacznie opuszczać cień i stopniowo coraz większa część jego tarczy będzie wydobywać się do obszaru półcienia. W momencie zewnętrznej styczności ze stożkiem cienia (czwarty kontakt), o godzinie 22^h07^m, zakończy się zaćmienie częściowe. Następnie do godziny 23^h16^m tarcza Księżyca pozostanie jeszcze w stożku półcienia Ziemi, który jednak będzie osłabiał blask naszego naturalnego satelity w stopniu prawie niedostrzegalnym.

Życzymy dobrej pogody i udanych obserwacji.

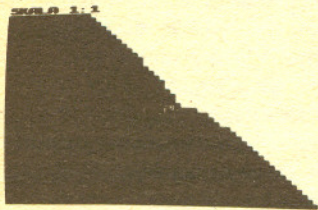
mgr Joanna UDALSKA

```

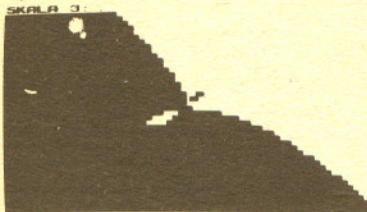
1 INPUT C
10 FOR X=-32 TO 31
20 FOR Y=-21 TO 20
30 IF 10*X*X+Y*(X-Y)+C+X*X+X*X+X*
   Y+Y+Y+Y+Y <= 0 THEN PLOT X+32,Y+21
40 NEXT Y
50 NEXT X
60 PRINT AT 0,0;"SKALA ":"C":"1"

```

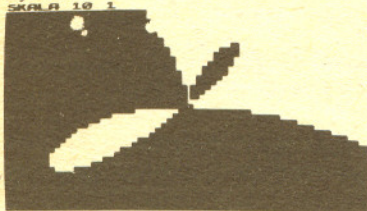
Rys. 1



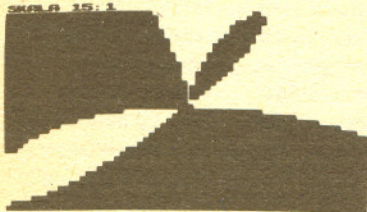
Rys. 2



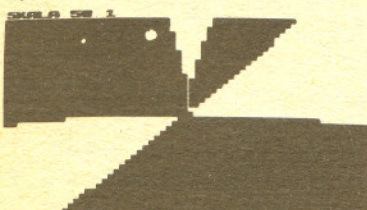
Rys. 3



Rys. 4



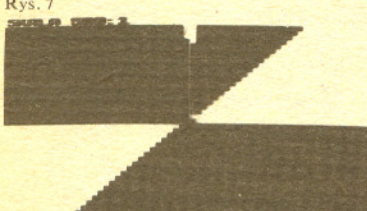
Rys. 5



Rys. 6



Rys. 7



Rys. 8



Dr hab. Jerzy JURKIEWICZ

Znacie mikrokomputer ZX81 Sinclaira? Można go kupić za kilkanaście dolarów. Niektórzy twierdzą, że jego „ploty” są zbyt grube, a więc obraz wytworzony przez to urządzenie na ekranie TV ma zbyt małą rozdzielczość dla subtelnych zastosowań graficznych. Otóż chcę Was przekonać że to nieprawda. Jeżeli jakiś szczegół oglądanego na ekranie rysunku jest za mały i niewyraźny, wystarczy powiększyć skalę! Oto przykład.

Interesuje nas krzywa K o równaniu

$$(*) \quad 10 X^2 Y(X - Y) + X^5 + Y^5 = 0,$$

to znaczy zbiór punktów płaszczyzny, których współrzędne spełniają to równanie. Komputer będzie szukał punktów tej krzywej próbując różne pary liczbowe X, Y . Ze względu na błędy zaokrągleń jest jednak mało prawdopodobne, aby otrzymać dokładnie równość (*) nawet, gdyby znalezione X i Y stanowiły dokładne rozwiązanie. Praktyczniej będzie zastąpić równanie (*) nierównością

$$(**) \quad 10 X^2 Y(X - Y) + X^5 + Y^5 \leq 0.$$

Jeżeli komputer zaczerni na ekranie „punkty”, których współrzędne spełniają tę nierówność, to krzywą K odnajdziemy jako *brzeg* zaczernionego obszaru.

(Dociekliwym Czytelnikom wyjaśniam, że wielomian występujący w równaniu (*) nie ma czynników wielokrotnych, więc ma on różne znaki, po obu stronach każdej gałęzi krzywej K .)

I jeszcze jedno. Ponieważ, jak się przekonamy, interesujący jest punkt $(0,0)$ naszej krzywej, więc aby lepiej zobaczyć jego otoczenie, przesuniemy rysunek na ekranie o wektor $(32,21)$ tak, aby punkt $(0,0)$ przeniósł się na środek ekranu. Tłumaczy to instrukcję „... THEN PLOT $X+32, Y+21$ ” w linii 30 programu (rys. 1).

Teraz zlecamy komputerowi sprawdzenie wszystkich punktów o współrzędnych całkowitych X, Y , gdzie $-32 \leq X \leq 31, -21 \leq Y \leq 20$. Jeżeli dla pewnego punktu (X, Y) zachodzi nierówność (**), to odpowiedni kwadracik ma być zaczerniony. Trochę to potrwa, bo do przebadania jest $64 \times 42 = 2688$ punktów. Program przedstawiony jest na rysunku 1. Występuje tam poza X i Y zmienna C , która oznaczać będzie skalę powiększenia, o czym dalej. Na razie nadajemy jej wartość 1, to znaczy po uruchomieniu programu czytujemy 1.

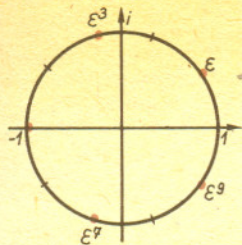
Po NEWLINE (czyli ENTER) na ekranie stopniowo ukazuje się zaczerniony obszar rysunku 2. Jak widać, w okolicy środka ekranu krzywa K , czyli brzeg zaczernionego obszaru ma jakąś osobliwość, prawie niewidoczną w tej skali. Po RUN dla uzyskania np. trzykrotnego powiększenia czytujemy 3. Powstanie obraz widoczny na rysunku 3. Oczywiście zwiększenie skali odbyło się kosztem zmniejszenia oglądanego fragmentu krzywej.

Powiększenie dziesięciokrotne jest ukazane na rysunku 4. Widać już wyraźnie, że krzywa ma dwa „liście” wyrastające ze środka ekranu. Nadal niejasne jest jednak, co dzieje się bezpośrednio nad środkiem. Kolejne powiększenia widoczne na rysunku 5, rysunku 6 i rysunku 7 wyjaśniają sprawę. W górę wyrasta szpiczasta gałąź krzywej, tak zwany cusp (czyt. kasp). Przy skali 500:1 (rys. 8) ukazany fragment krzywej K sprowadza się, praktycznie biorąc, do trzech prostych. Dalsze powiększenia nic już nie wniosą. Rysunek 8 można interpretować następująco, posługując się równaniem (*). Dla X i Y bardzo bliskich zera (w porównaniu z 1) składnik $X^5 + Y^5$, który jest stopnia 5, jest bardzo mały w porównaniu ze składnikiem $10 \cdot X^2 Y(X - Y)$, mającym stopień 4. Zatem w pobliżu punktu $(0,0)$ krzywa K opisana jest dość dokładnie równaniem

$$X^2 Y(X - Y) = 0,$$

a więc omalże pokrywa się z sumą prostych $X = 0, Y = 0$ i $X - Y = 0$. Istotnie widzimy, że te właśnie proste stanowią brzeg czarnego obszaru w pobliżu punktu $(0,0)$. Nie zapominajmy, że ten punkt umieściliśmy na środku ekranu! Opisane linie nazywamy prostymi stycznymi do gałęzi krzywej K w punkcie $(0,0)$.

Wróćmy do rysunku 2, który przedstawia krzywą w skali 1:1 (oczywiście dla danego ekranu). Dla dalekich od zera (w porównaniu z 1) liczb X i Y składnik $X^5 + Y^5$ jest duży w porównaniu ze składnikiem $10 X^2 Y(X - Y)$, więc równanie (*) jest dobrze przybliżane równaniem $X^5 + Y^5 = 0$. W dziedzinie liczb rzeczywistych opisuje ono prostą $X + Y = 0$, bo $X^5 = -Y^5 = (-Y)^5 \Leftrightarrow X = -Y$.



Spróbuj, Czytelniku, obejrzeć krzywą o równaniu

$$X^2 - Y^2 + Y^4 = 0.$$

Inne ciekawe rysunki można otrzymać rozpatrując równania postaci

$$(X + A_1 Y)^{n_1} \cdot (X + A_2 Y)^{n_2} \cdot \dots \cdot (X + A_k Y)^{n_k} + w(X, Y) = 0,$$

gdzie $w(X, Y)$ jest wielomianem jednorodnym stopnia większego niż $n_1 + n_2 + \dots + n_k$, tzn.

$$w(X, Y) = B_0 X^n + B_1 X^{n-1} Y + \dots + B_{n-1} X Y^{n-1} + B_n Y^n,$$

gdzie $n > n_1 + \dots + n_k$.

Tę właśnie prostą widzimy (w przybliżeniu) na rysunku. Wyznacza ona tak zwany punkt w nieskończoności albo punkt niewłaściwy krzywej K . Gdyby ekran ukazywał rozwiązania nierzeczywiste (zespolone), zobaczylibyśmy jeszcze cztery pozostałe punkty w nieskończoności, odpowiadające „prostym” $X = \epsilon^k \cdot Y$, gdzie $\epsilon = \cos \frac{2\pi}{10} + i \sin \frac{2\pi}{10}$, a $k = 1, 3, 7$ lub 9 (rys. 9).

Widzimy więc, że oglądając krzywą K w różnej skali możemy zbadać ją z lokalnego i z globalnego punktu widzenia.

Wróćmy do parametru C . Niech C będzie liczbą rzeczywistą różną od zera i niech K' oznacza krzywą powstałą z K przez C -krotne powiększenie. Mamy

$$(X, Y) \in K' \Leftrightarrow (X/C, Y/C) \in K,$$

a zatem krzywa K' opisana jest równaniem

$$10(X/C)^2(Y/C)(X/C - Y/C) + (X/C)^5 + (Y/C)^5 = 0.$$

Mnożąc obie strony przez C^5 mamy

$$10 X^2 Y (X - Y) \cdot C + X^5 + Y^5 = 0.$$

Język BASIC ma co prawda operację potęgowania, ale jest ona poprawna tylko dla dodatnich argumentów. Skomplikowałoby to program na tyle, że praktyczniej będzie w naszym przypadku napisać $X * X * X * X * X$ zamiast $X ** 5$ itd. Wyjaśnia to postać najważniejszej, 30 linii programu na rysunku 1.

Czego nie potrafi komputer

Mgr Jarosław DEMINET

Czy komputer potrafi wszystko?

Wiadomo, że obecnie wiele problemów przekracza możliwości współczesnych komputerów, np. gra w szachy (na poziomie mistrza świata) albo niezawodna prognoza pogody. Nie ma jednak podstaw do przypuszczeń, że tych akurat problemów nie da się rozwiązać kiedyś w przyszłości. Czy jednak istnieją problemy, których żaden komputer nigdy nie rozwiąże? Okazuje się, że tak. Niektóre z nich są zresztą bardzo proste. Oto przykład.

Wiadomo, że każdy niebanalny program komputerowy zawiera pętle, tzn. ciągi instrukcji wykonywane wielokrotnie. Kontynuowanie obliczeń w pętli przebiega do momentu spełnienia określonego warunku, np. w algorytmie szybkiego sortowania (*Delta* 9/1985) zewnętrzna pętla wykonywała się, dopóki $i < j$, a zatem kończyła się, gdy $i \geq j$. Zarówno i , jak i j były zmieniane wewnątrz pętli, przy czym i było zwiększane, a j zmniejszane. Można dowiedzieć, że po skończonej liczbie wykonania pętli rzeczywiście obie zmienne przyjmą takie wartości, że $i \geq j$. Co jednak byłoby, gdybyśmy pisząc algorytm zrobili jakiś błąd i gdyby dla pewnych danych wejściowych warunek $i \geq j$ nigdy nie był spełniony? Powiedzielibyśmy wówczas, że program się zapętlił — pewien jego fragment wykonywałby się nieskończenie długo i wymagałby przerwania z zewnątrz. Oczywiście sytuacja taka jest przykra i każdy chce jej uniknąć. Własność programu polegająca na tym, że program zawsze zakończy swoją pracę, nazywamy własnością stopu. Na ogół potrafimy udowodnić „ręcznie” własność stopu dla swojego programu. Może wobec tego można napisać program komputerowy, który potrafiłby automatycznie dowodzić własność stopu lub jej brak dla dowolnego programu? Okazuje się, że nie, a najprościej tego dowiedzieć przez sprowadzenie do sprzeczności.

Przypuśćmy, że mamy procedurę WŁASNOŚĆ—STOPU (X), dla której parametrem X jest tekst dowolnej procedury, zapisanej w jakimś języku programowania. Procedura daje w wyniku

wartość logiczną *prawda*, gdy procedura X ma własność stopu, zaś *falsz*, gdy tej własności nie posiada. Oczywiście dysponujemy tekstem procedury WŁASNOŚĆ—STOPU. Teraz możemy napisać bezparametrową procedurę:

ZAGADKA:

dopóki WŁASNOŚĆ—STOPU (ZAGADKA) wykonuj
początek
 nic nie rób
koniec

Tekst tej prostej procedury przekazujemy jako parametr procedurze WŁASNOŚĆ—STOPU. Jaki będzie wynik?

Załóżmy, że WŁASNOŚĆ—STOPU (ZAGADKA) jest prawdą. Ale to znaczy, że warunek pętli w procedurze ZAGADKA będzie zawsze spełniony, a zatem pętla będzie się wykonywać nieskończenie i ZAGADKA nie ma własności stopu.

Załóżmy, że WŁASNOŚĆ—STOPU (ZAGADKA) jest fałszem. To znaczy, że pętla w procedurze ZAGADKA nie wykona się ani razu i procedura od razu zakończy się, a zatem ma ona własność stopu.

W obu przypadkach dochodzimy do sprzeczności. Oznacza to, że procedura WŁASNOŚĆ—STOPU o szukanych własnościach nie istnieje. Sprzeczność zniknie, gdy założymy, że procedura WŁASNOŚĆ—STOPU sama nie ma własności stopu, tzn. że dla niektórych wartości swojego parametru (dla niektórych procedur) nie daje w ogóle odpowiedzi w skończonym czasie. Ale to już nie to...

A więc nigdy nie da się zaprogramować komputera tak, aby określał, czy dowolny program zakończy swoją pracę. Mówimy, że własność stopu *nie jest rozstrzygalna*. Okazuje się zresztą, że wiele innych własności programów nie jest rozstrzygalnych. Nie istnieje np. algorytm znajdujący zbędne instrukcje i zmienne (zbędne — tzn. takie, które zawsze będą ominięte przez program). Można bowiem dowiedzieć, że gdyby taki algorytm istniał, to pozwoliłby on również na dowodzenie własności stopu.

A dla programistów wynika z tego morał: żaden komputer nie sprawdzi naszych programów całkowicie i do końca. Musimy błysnąć intuicją i sami znajdować dowody własności stopu naszych programów.

Rozwiązanie zadania F 204. Uderzenie pioruna powoduje przepływ prądu w rurce i przewodniku w tym samym kierunku. Prąd o natężeniu I w przewodniku jest źródłem pola magnetycznego, które w odległości b od przewodnika ma wartość $B = \mu_0 I / 2\pi b$ (μ_0 — przenikalność magnetyczna próżni). Ponieważ w rurce również płynie prąd, na każdy jej fragment działa w tym polu siła skierowana do osi. Jeśli założymy, że natężenia prądu w rurce i przewodniku są równe, to ciśnienie działające na rurkę będzie miało wartość $p = B \cdot I / 2\pi b = \mu_0 I^2 / 4\pi^2 b^2$. Zgniecenie nastąpi, jeśli $p \geq p_{kr} = 30$ kPa. Stąd otrzymujemy warunek

$$\frac{\mu_0}{4\pi^2 b^2} \left(\frac{Q}{t} \right)^2 \geq p_{kr},$$

czyli

$$Q \geq 2\pi b t \sqrt{\frac{p_{kr}}{\mu_0}}.$$

Po podstawieniu podanych wartości i $\mu_0 = 4\pi \cdot 10^{-7} \text{ V} \cdot \text{s/A} \cdot \text{m}$ otrzymujemy $Q \geq 10 \text{ C}$, czyli prąd o natężeniu większym niż 10 kA.

Mgr Jarosław DEMINET

Jeszcze do niedawna podstawową formą komunikowania się człowieka z komputerem był mniej lub bardziej ożywiony dialog: człowiek wprowadzał kolejne linie tekstu z klawiatury (mogły to być rozkazy do natychmiastowego wykonania lub linie programu), a komputer komentował je drukując odpowiedzi na papierze lub wyświetlając na ekranie. Wiadomo jednak, że nie jest to szybka i niezawodna forma komunikacji. Wyobraźmy sobie, że kierowca miałby prowadzić samochód posługując się klawiaturą i ekranem. Na ekranie pojawiałyby się np. tekst „Dojeżdżasz do skrzyżowania” albo „Przed tobą hamuje samochód”, a kierowca musiałby zareagować napisaniem „Hamuj” albo „W lewo”. Podróż skończyłaby się zapewne bardzo szybko. Oczywiście można by wprowadzić jednoliterowe skróty komend, ale wtedy łatwo byłoby o błędy wynikające z zapomnienia, czy S oznacza „Włącz ssanie”, czy też „Stop”. Przeciętnemu człowiekowi znacznie lepiej od pisania na maszynie wychodzi pokazywanie palcem, naciskanie przycisków (jeśli jest ich niezbyt dużo i są wyraźnie oznakowane) albo poruszanie dźwignią.

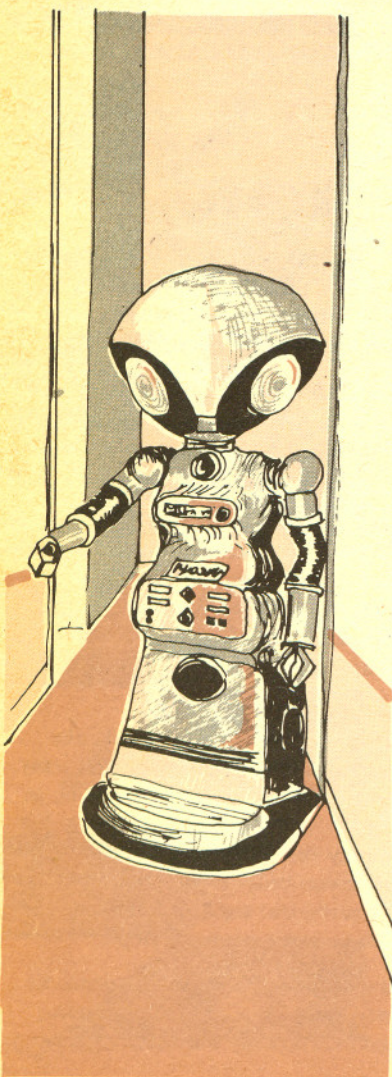
Początkowo, gdy komputery były bardzo drogie, a dostęp do nich mieli tylko wybrani „kapłani kultu”, na ogół nie zwracano zbytniej uwagi na ułatwienie kontaktu człowieka z maszyną. Wyjątkami były zastosowania specjalne — np. komputer sterujący radarem wyświetlał informację w postaci graficznej na wielkim ekranie i przyjmował polecenia za pośrednictwem kilku różnokolorowych przycisków.

W miarę jak krąg użytkowników komputerów powiększał się, zaczęto doceniać znaczenie graficznej formy przedstawiania informacji przez komputer oraz wprowadzania komend poprzez wskazywanie ustalonych obszarów na ekranie monitora. Początkowo bardzo popularne były tzw. pióra świetlne. Przy ich konstrukcji skorzystano z faktu, że obraz na ekranie nie powstaje od razu w całości, lecz że jest tworzony punkt po punkcie i linia po linii. Pióro świetlne zawiera fotokomórkę, którą można skierować na dowolny punkt ekranu. Zareaguje ona na rozświetlenie tego właśnie punktu, a wówczas można sprawdzić, jaki element rysunku jest właśnie wyświetlany (np. która linia i który punkt w linii). Pióro świetlne konstrukcyjnie jest dość proste, ma jednak także wady. Po pierwsze, wymaga, aby użytkownik znajdował się na tyle blisko monitora, żeby mógł go dotykać ręką. Po drugie, komputer jest w stanie określić położenie pióra tylko wówczas, gdy wskazuje ono świecący element ekranu — nie można piórem „rysować” na ciemnym tle. Na ogół rozwiązuje się ten problem rozświetlając co jakiś czas (np. co 2 sekundy) wszystkie punkty ekranu, co trwa 1/25 s. Taki migający ekran nie zawsze jednak jest do przyjęcia.

Inne rozwiązanie polega na zainstalowaniu na dwóch prostopadłych bokach ekranu zestawu diod świecących w podczerwieni równoległe do powierzchni ekranu. Na przeciwległych bokach ekranu są fotokomórki. Dotknięcie ekranu palcem lub ołówkiem powoduje przecięcie promieni świetlnych odpowiadających współrzędnym dotkniętego punktu. Można w ten sposób wybierać dowolny punkt ekranu, ale koszt całego urządzenia jest spory, a poza tym, podobnie jak pióro świetlne, wymaga ono, aby użytkownik znajdował się blisko ekranu.

W zastosowaniach projektowych często używa się specjalnych stołów kreślarskich. Pozwalają one bardzo precyzyjnie wprowadzać położenie punktów do komputera w postaci cyfrowej, ale ich koszt przekracza często koszt całej reszty komputera.

Ostatnio największą popularność zyskała tzw. myszka. Jest to małe pudełko, mieszczące się wygodnie w dłoni. Myszka może się poruszać po stole, ma ogonek (czyli przewód) łączący ją z komputerem oraz rozmaity liczbę oczek (przycisków), które można naciskać. W środku znajduje się plastikowa kulka, oparta o dwa prostopadłe kółka. Ruch myszki po stole powoduje obrót kulki, rozkładany na prostopadłe składowe przez oba kółka. Informacja o ruchu kółek jest przekazywana do komputera. Jak z tego wynika, komputer nie wie, gdzie w danym momencie jest myszka — wie tylko, w którą stronę została ona przesunięta po stole. Funkcjonowanie myszki opiera się o sprzężenie zwrotne między człowiekiem a komputerem. Komputer wyświetla w pewnym miejscu ekranu tzw. kursor, odpowiadający wyimaginowanemu położeniu myszki. Ruch myszki po stole powoduje przesuwanie kursora na ekranie. Człowiek chcąc przesunąć kursor do wybranego punktu na ekranie musi więc wodzić myszką w odpowiednią stronę. Oczywiście bezwzględne położenie myszki jest bez znaczenia — podniesienie jej i postawienie w innym miejscu nie zmienia położenia kursora na ekranie.





Rozwiązanie zadania M 447. Niech $\xi_{n+1} = S_{n+1} - S_n$; zdarzenia postaci $\{S_n = k\}$ i $\{\xi_{n+1} = j\}$ są niezależne. Wobec tego

$$\begin{aligned}
 E\left(\frac{q}{p}\right)^{S_{n+1}} &= \sum_{k=-(n+1)}^{n+1} \left(\frac{q}{p}\right)^k P(S_{n+1} = k) = \\
 &= \sum_{k=-n}^n \left(\frac{q}{p}\right)^{k+1} P(S_n = k, \xi_{n+1} = 1) + \\
 &+ \sum_{k=-n}^n \left(\frac{q}{p}\right)^{k-1} P(S_n = k, \xi_{n+1} = -1) = \\
 &= \sum_{k=-n}^n \left(\frac{q}{p}\right)^k P(S_n = k) \cdot \frac{q}{p} \cdot p + \\
 &+ \sum_{k=-n}^n \left(\frac{q}{p}\right)^k P(S_n = k) \cdot \frac{p}{q} \cdot q = E\left(\frac{q}{p}\right)^{S_n}.
 \end{aligned}$$

W takim razie dla każdego n mamy

$$E\left(\frac{q}{p}\right)^{S_n} = E\left(\frac{q}{p}\right)^{S_0} = 1.$$

Równocześnie z doskonaleniem sprzętu wymyślano nowe narzędzia dla programistów. Dawniej cały ekran monitora należał do jednego programu, który mógł wypisywać na nim swoje informacje, na ogół określając bezpośrednio współrzędne tekstów czy rysunków. Zmiana układu informacji na ekranie wymagała sporych przeróbek programu. Gdy użytkownik nieopatrznie uruchomił dwa programy wypisujące informacje na ekran, to poszczególne fragmenty wyników obu programów mieszały się dając w efekcie zupełny bałagan. Obecnie coraz częściej stosuje się podział ekranu między tzw. okna. Każdy program może wypisywać informacje do swojego okna, które w rzeczywistości stanowi tylko część ekranu. Różne okna mogą się częściowo lub całkowicie zasłaniać, tak jak kartki papieru rozłożone na biurku. Komputer pamięta, co ostatnio zapisano w każdym oknie, nawet jeśli jest ono zasłonięte. Użytkownik może zażądać odsłonięcia okna i wówczas zapisana w nim informacja znów staje się widoczna. Przekładanie okien z miejsca na miejsce odbywa się za pomocą myszki: jeśli doprowadzi się kursor do częściowo przysłoniętego okna i naciśnie jedno z oczek myszki, to wskazane okno wędruje „na powierzchnię” stosu okien. Natomiast naciśnięcie innego oczka powoduje, że okno przykleja się do kursora i może zostać przeciągnięte w dowolny rejon ekranu. Dotknięcie kursorem jednego z rogów okna powoduje jego zwiększenie lub zmniejszenie. Istotne jest to, że wszystkie operacje związane z oknami nie wymagają żadnego wysiłku od programisty — zajmuje się nimi specjalny program, zwany GEM (ang. Graphics Environment Manager, czyli program zarządzania środowiskiem graficznym; słowo „gem” oznacza także szlachetny kamień).

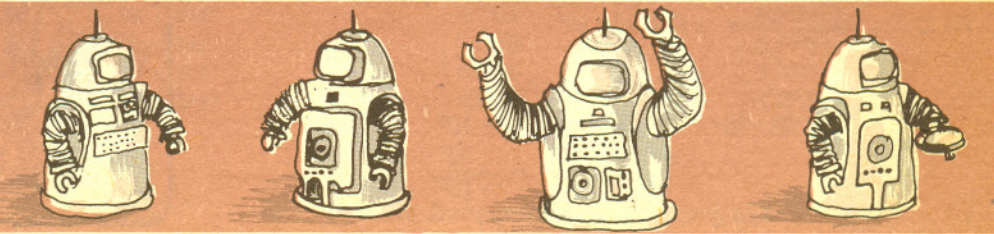
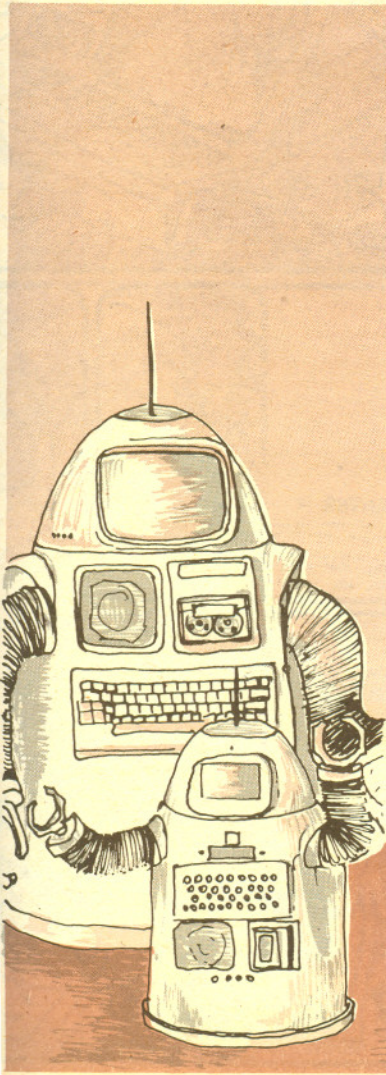
Może ktoś zapytać, po co w ogóle możliwość równoczesnego oglądania na ekranie kilku okien zawierających informacje od różnych programów. Przykładów zastosowań jest wiele. Przy uruchamianiu nowego programu programista chce oglądać to, co jego program normalnie wypisywałby na ekran, żeby ocenić działanie programu. Gdy jednak program pracuje błędnie, dobrze jest zerknąć na jego treść, żeby zlokalizować błąd. Bez okienek programista musiał mieć tekst programu wydrukowany zawczasu na papierze. Często przy znajdowaniu błędu jest pomocna znajomość stanu programu, np. tego, jakie są wartości rozmaitych zmiennych. Specjalny program nadzorujący może nam to wypisywać w dodatkowym okienku.

Inny przykład dotyczy komputera zainstalowanego w biurze. Podstawowe okno, stanowiące na ogół tło dla innych, zawiera zegar i kalendarz oraz notes do zapisywania krótkich wiadomości i zleceń. Sekretarka korzystająca z programu przygotowującego listy może w razie potrzeby zamknąć chwilowo okno tego programu (usunąć je z ekranu), aby zanotować np. treść rozmowy telefonicznej. Może też otworzyć okno zawierające numery telefonów, wybrać (myszką) jeden z nich i polecić, aby komputer „wykreślił” go. Po odbytej rozmowie może ponownie otworzyć okno zawierające przygotowywany list.

Myszka służy nie tylko do rysowania i do przesuwania okien. Wiele programów wypisuje listę dozwolonych komend na marginesie swego okna. Użytkownik, aby wydać komendę, nie musi mozolnie wprowadzać jej nazwy litera po literze. Wystarczy przesunąć kursor tak, aby dotknął komendy i nacisnąć odpowiednie oczko myszki.

Popatrzmy teraz, jak wygląda posługiwanie się myszką przy poprawianiu tekstu. Fragment tekstu pojawia się w okienku. Użytkownik chce usunąć jedno zdanie. Przesuwa myszkę tak, aby kursor dotknął zdania i naciska jedno z oczek. Zdanie zostaje podkreślone (albo zaczyna migać). Teraz można wybrać kursorem komendę „Usuń” (znajdącą się na marginesie okienka) i ponownie nacisnąć oczko. Zdanie zniknie, a reszta tekstu zostanie przesunięta tak, aby zapisać puste miejsce. Aby przesunąć okienko na inną część tekstu, należy wybrać kursorem strzałkę znajdującą się na marginesie okienka (i nacisnąć oczko). Prawda, że proste?

Wszystkie opisane tu mechanizmy i udogodnienia powstały po to, aby maksymalnie ułatwić użytkownikowi komunikację z komputerem. Wydaje się, że cel ten osiągnięto — w ciągu kilku godzin każdy, od sekretarki i gospodyni domowej do dyrektora, może się nauczyć sprawnie korzystać z możliwości sprzętu i oprogramowania.



5 mała delta

Czy dodawanie jest przemienne?

W szkole uczy się różnych własności dodawania. Jest ono przemienne, łączne, itd. Warto wiedzieć, na ile te szkolne wiadomości dotyczą tegoż dodawania, ale wykonywanego na kalkulatorach czy też komputerach. Okazuje się, że jedyną własnością dodawania, która pozostaje prawdziwa dla dodawania „maszynowego”, jest przemienność.

W komputerach i nieco lepszych typach kalkulatorów liczba przedstawiana jest w postaci zapisu zmiennopozycyjnego. Przypuśćmy, że do zapisu liczby możemy użyć tylko sześciu cyfr. Dobrym sposobem jest wykorzystanie czterech miejsc do zapisu czterech cyfr znaczących liczby, a pozostałych dwóch miejsc do zapisu miejsca położenia przecinka.

Tak więc

$$\begin{aligned} 5765000 &= 0,5765 \cdot 10^7 \\ -28,83 &= -0,2883 \cdot 10^2 \\ 0,00732 &= 0,7320 \cdot 10^{-2} \text{ itd.} \end{aligned}$$

Zamiast podstawy 10 można stosować inne (w komputerach — podstawa 2); liczba cyfr znaczących może być inna (na ogół większa), ale zasada zapisu jest taka sama.

Jak dodawać takie liczby używając sumatora, w którym można zapisywać np. sześć cyfr znaczących? Najlepiej popatrzeć na przykłady.

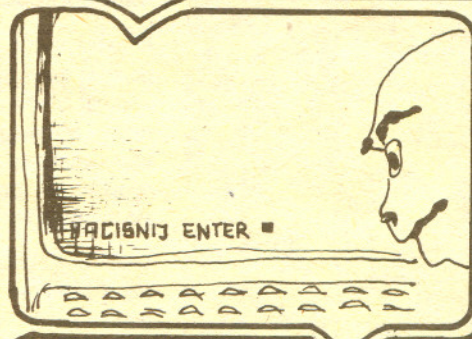
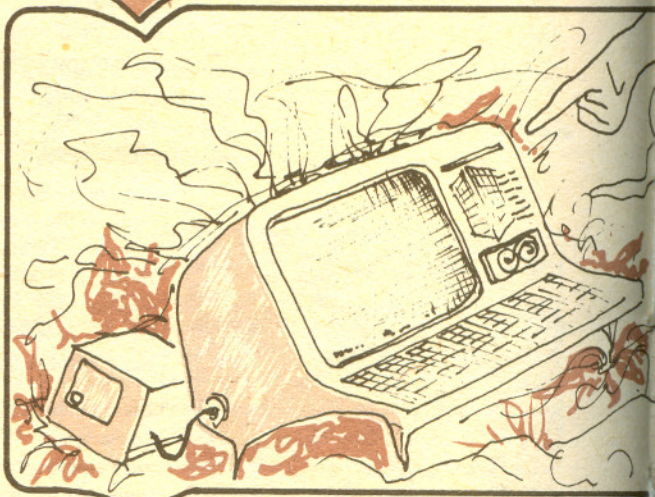
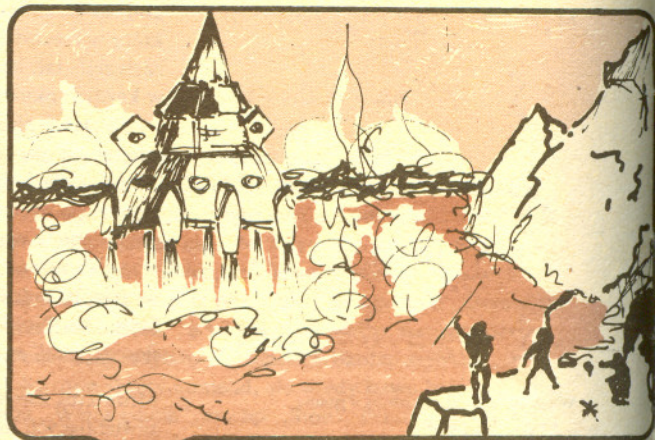
Niech $a = 0,5768 \cdot 10^3$, $b = 0,2315 \cdot 10^5$,
 $c = 0,9785 \cdot 10^{-3}$.

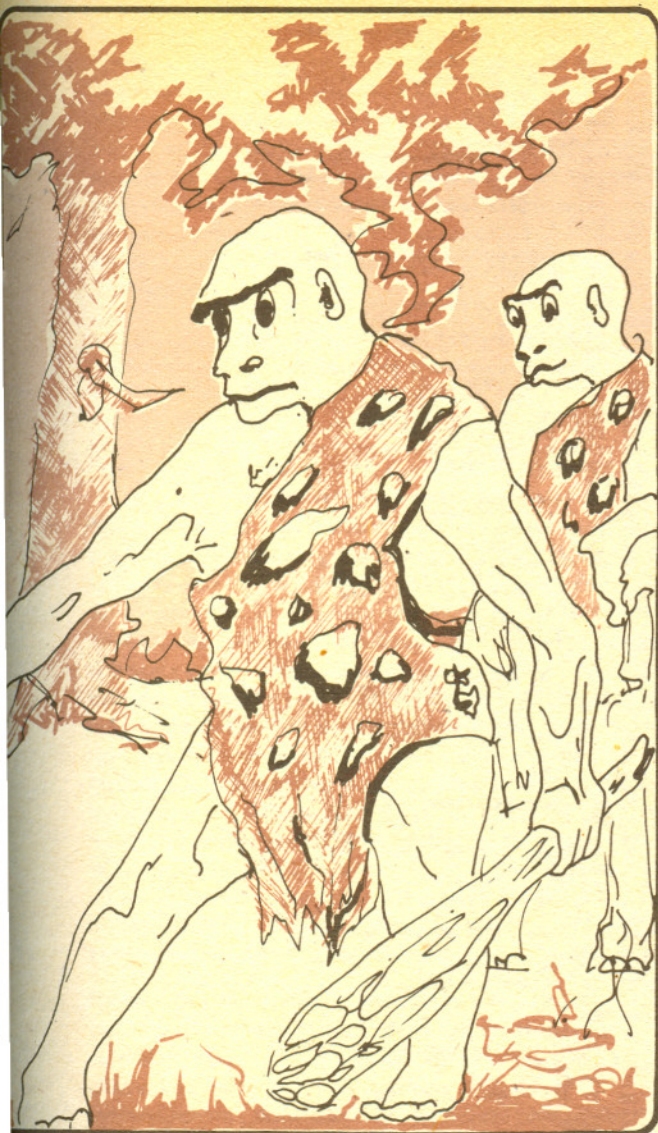
$$\begin{array}{r} 0,5768 \cdot 10^3 = 0,005768 \cdot 10^5 \\ + 0,231500 \cdot 10^5 \\ \hline 0,237268 \cdot 10^5 \end{array}$$

Po zaokrągleniu $b+a = 0,2373 \cdot 10^5$.

$$\begin{array}{r} 0,9785 \cdot 10^{-3} = 0,0000009785 \cdot 10^5 \\ + 0,231500 \cdot 10^5 \\ \hline 0,231500 \cdot 10^5 \end{array}$$

Po zaokrągleniu $c+a = 0,2315 \cdot 10^5$.





Taka jest zasada dodawania liczb w kalkulatorach i komputerach. Oczywiście, takie dodawanie jest przemienne. A jak z innymi własnościami?

Drugi przykład pokazuje, że z równości $x + y = x$ nie musi wynikać, iż $y = 0$.

Łączności też nie ma. Jeśli $a = 0,5571 \cdot 10^{-5}$, $b = 0,1523 \cdot 10^{-5}$, $c = 0,2313 \cdot 10^{-2}$, to $(a+b)+c = 0,2320 \cdot 10^{-2}$, ale $a+(b+c) = 0,2321 \cdot 10^{-2}$.

Może się też zdarzyć, że $a+b = 0$, ale $a \neq -b$. Gdy $a = 0,2517 \cdot 10^{-98}$ i $b = -0,2516 \cdot 10^{-98}$, to $a+b = 0,0001 \cdot 10^{-98} = 0,1 \cdot 10^{-101}$. Taki wynik zwykle jest uznawany przez maszynę za równy 0.

Sprawdzenie, co się dzieje z innymi własnościami dodawania, pozostawiamy Czytelnikowi.

O specyfice dodawania przybliżonego warto też pamiętać, gdy mamy dodawać dużo liczb różniących się o kilka rzędów wielkości. Sumowanie od najmniejszej do największej z reguły daje dokładniejszy wynik niż sumowanie od największej do najmniejszej. Na przykład chcemy dodać naszym sumatorem dziesięć tysięcy liczb: jedna jest równa $0,1 \cdot 10^5$, a pozostałe równe $0,1 \cdot 10^1$.

Dodawanie $0,1 \cdot 10^5$ i $0,1 \cdot 10^1$ daje $0,1 \cdot 10^5$, a więc rozpoczynając sumowanie od największej otrzymujemy wynik równy $0,1 \cdot 10^5$. Tymczasem sumując od najmniejszej dostajemy po 9998 dodawaniach liczbę $0,9999 \cdot 10^4$ i po dodaniu $0,1 \cdot 10^5$ ostatecznym wynikiem jest $0,2 \cdot 10^5$.

W tabelce przedstawione są wyniki sumowania liczb

$$\frac{1}{1 \cdot 2 \cdot 3}, \frac{1}{2 \cdot 3 \cdot 4}, \frac{1}{3 \cdot 4 \cdot 5}, \dots, \frac{1}{n(n+1)(n+2)}$$

obydwoma sposobami dla różnych wartości n .

(Mówiąc dokładniej: sumujemy — zamiast powyższych liczb — ich przybliżenia z dokładnością do czterech cyfr znaczących; po każdym dodawaniu wynik zaokrąglamy do czterech cyfr znaczących.) W ostatniej rubryce podany jest wynik prawidłowy (tzn. $\frac{1}{4} - \frac{1}{2(n+1)(n+2)}$) z dokładnością do sześciu cyfr znaczących.

n	$\frac{1}{1 \cdot 2 \cdot 3} + \dots + \frac{1}{n(n+1)(n+2)}$	$\frac{1}{n(n+1)(n+2)} + \dots + \frac{1}{1 \cdot 2 \cdot 3}$	$\frac{1}{4} - \frac{1}{2(n+1)(n+2)}$
3	0,2382	0,2381	0,238095
10	0,2464	0,2462	0,246212
20	0,2491	0,2489	0,248918
25	0,2495	0,2493	0,249288
30	0,2497	0,2495	0,249496
40	0,2497	0,2497	0,249710
60	0,2497	0,2499	0,249868
80	0,2497	0,2499	0,249925
100	0,2497	0,2500	0,249951
300	0,2497	0,2500	0,249994
1000	0,2497	0,2500	0,250000



Małą Deltę przygotował Jerzy RYLL

Metoda Monte Carlo a ruchu Browna

Dr hab. Ryszard KUTNER

W ostatnich kilkunastu latach, wraz z prawdziwą inwazją nie tylko w wielu gałęziach nauki różnego typu komputerów najnowszych generacji, nastąpił szybki rozwój m.in. numerycznych metod symulacji komputerowej. Znalazły one zastosowanie głównie w dziedzinie fizyki materii skondensowanej, w teorii pola, w fizyce stosowanej czy też w dziedzinie chemii fizycznej, przyczyniając się w istotny sposób do ich rozwoju. Do perfekcji doprowadzono metody: deterministyczną dynamiki molekularnej, deterministyczno-stochastyczną dynamiki Langevina, a przede wszystkim, szybszą od innych stochastyczną metodę Monte Carlo. Szczególnie spektakularnym, a zarazem poglądowym przykładem zastosowania tej ostatniej metody może być symulowanie komputerowe ruchów Browna.

Pogłębioną fizyczną analizę ruchów Browna Czytelnik może znaleźć np. w dwuczęściowym artykule B. Cichockiego pt. „Ruchy Browna”, *Delta* nr 4 i 5 (1983).

Zrozumienie istoty zjawiska ruchów Browna nastąpiło na początku bieżącego stulecia i związane jest z nazwiskiem Alberta Einsteina, Paula Langevina, a przede wszystkim Mariana Smoluchowskiego, któremu zawdzięczamy wyjaśnienie mechanizmu tego zjawiska. Porównanie przewidywań teorii z doświadczeniem pozwoliło Jeanowi Perrin na wyznaczenie m.in. liczby Avogadra, a zatem bezwzględnych mas atomowych i stanowiło przekonujący dowód realności tzw. hipotezy atomistycznej.

Charakterystyczna własność ruchów Browna to występująca nieustannie nieregularna zmiana położenia cząsteczki o rozmiarach rzędu 10^{-4} cm, zawieszona w cieczy lub w gazie, wywołana przypadkowymi potrąceniami ze strony otaczających ją znacznie mniejszych cząsteczek ośrodka. Inaczej mówiąc — na cząsteczkę zawieszoną działa fluktuująca siła spowodowana chaotycznymi, nieskompensowanymi uderzeniami cząsteczek ośrodka. Wynik pojedynczego, całkowicie przypadkowego zderzenia jest bardzo mały (w skali makroskopowej), jednak sumarycznym efektem dużej liczby tych zderzeń może być, obserwowane przez mikroskop nawet o niewielkim powiększeniu, znaczne wypadkowe przesunięcie przypadkowe cząstki zawiesiny. Przejawiająca się w postaci zygzakowatej trajektorii błądzącej cząstki zawiesiny, leży u podstaw symulacji ruchów Browna metodą Monte Carlo.

Metoda Monte Carlo została omówiona np. w sześcioczęściowym artykule R. Zielińskiego pod wspólnym tytułem: „Metody Monte Carlo”, *Delta* nr 9—11 (1975) oraz *Delta* nr 1—3 (1976).

Metoda ta pozwala na zbudowanie modelu numerycznego — jest on zrealizowany w postaci (odtworzonego na drodze numerycznej) procesu stochastycznego Markowa, zwanego w tej sytuacji dokładniej błądzeniem przypadkowym. Najistotniejszym elementem tego, w istocie rzeczy uproszczonego modelu, jest przypadkowe wybieranie przez komputer kierunku każdego, kolejnego elementarnego przesunięcia Δr_j cząstki ($j = 1, \dots, N$, gdzie N jest całkowitą liczbą elementarnych przemieszczeń).

Liczby generowane przez funkcję RND mikrokomputera nie są przypadkowe „sensu stricto” — są to liczby pseudolosowe, chyba że komputer korzysta z tzw. generatora szumowego, czyli, najogólniej mówiąc, przetwornika fluktuacji prądowych na sygnały binarne, rozumiane przez komputer właśnie jako liczby przypadkowe.

```
10 REM Ruch Browna Pojedynczo#90
20 REM atomu w zawieszynie.
30 REM Program wyznacza m.in <math>r^2(t)</math>
40 REM w eksperymentach komputerowych.
50 REM
60 REM Macierze i dane
70 DIM u(50): DIM v(50): DIM d(50): DIM r(50)
80 READ tk,a,s,p
90 DATA 40,6,100,50
100 REM
110 REM Tabela
120 PRINT "Czas eksperymentu=";tk;" [KMC]"
130 PRINT "-----"
140 PRINT : PRINT
150 PRINT "<math>\langle \text{Dalej dowolny klawisz} \rangle</math>"
160 PAUSE 0: CLS
170 REM
180 REM Punkt startu oznaczono przez "0"
190 LET a0=120: LET b0=76
200 REM
210 REM Segment dyfuzyjny
220 FOR j=1 TO s
230 PRINT "nr exper.=",j;" ; " ; "liczba exper.=",s
240 REM
250 REM Rysuje Pojemnik i Punkt startowy
260 PLOT 0,0: DRAW 0,151
270 PLOT 0,151: DRAW 255,0
280 PLOT 255,151: DRAW 0,-151
290 PLOT 255,0: DRAW -255,0
300 CIRCLE a0,b0,4
310 REM *****
320 REM
330 REM Pojedynczy eksperyment statystyczny
340 LET x0=a0: LET y0=b0
350 FOR t=1 TO tk
360 LET f=2*PI*RND
370 LET dx=a*COS(f): LET x1=x0+dx
380 LET dy=a*SIN(f): LET y1=y0+dy
390 IF x1>=255 THEN GO TO 410
400 GO TO 420
410 LET x1=2*255-x1: LET dx=x1-x0
420 IF y1>=151 THEN GO TO 440
430 GO TO 450
440 LET y1=2*151-y1: LET dy=y1-y0
450 IF x1<=0 THEN GO TO 470
460 GO TO 480
470 LET x1=-x1: LET dx=x1-x0
480 IF y1<=0 THEN GO TO 500
490 GO TO 550
500 LET y1=-y1
510 LET dy=y1-y0
520 REM *****
530 REM
540 REM Rysuje kolejne pozycje atomu
550 PLOT x0,y0: DRAW dx,dy
560 REM
570 REM Oblicza <math>r^2(t)</math>
580 LET u(t)=x1: LET v(t)=y1
590 LET d(t)=(u(t)-a0)*(u(t)-a0)+(v(t)-b0)*(v(t)-b0)
600 LET x0=x1: LET y0=y1
610 NEXT t
620 FOR t=1 TO tk
630 LET r(t)=r(t)+d(t)/s
640 NEXT t
650 REM
660 IF j=2 THEN GO TO 690
670 PRINT "<math>\langle \text{Dalej dowolny klawisz} \rangle</math>"
680 PAUSE 0: CLS
690 PAUSE p
700 IF j=s THEN GO TO 720
710 CLS
720 NEXT j
730 PRINT "<math>\langle \text{Dalej dowolny klawisz} \rangle</math>"
740 PAUSE 0: CLS
750 REM
760 REM Tabela
770 PRINT "t"; " " ; "<math>r^2(t)</math>"
780 PRINT "-----"
790 PRINT
800 FOR t=1 TO tk
810 PRINT t;" " ; " ; r(t)
820 NEXT t
830 PRINT
840 PRINT "Teoretyczny wsp. dyfuzji D=";a*a/4;"
      Ejednostki eksp. komputerowe, tzn. 1kMC=1]"
850 PRINT
860 PRINT "<math>\langle \text{Dalej dowolny klawisz} \rangle</math>"
870 PAUSE 0: CLS
880 REM
890 REM Konstrukcja wyskalowanych osi x-y
900 PRINT " 2^ 3"
910 PRINT "r x10": PRINT : PRINT : PRINT " 2": PRINT :
      PRINT : PRINT : PRINT : PRINT : PRINT " 1":
      PRINT : PRINT : PRINT : PRINT : PRINT : PRINT : PRINT
920 PRINT TAB 29;">"
930 PRINT TAB 7;"2";TAB 12;"4";TAB 17;"6";TAB 22;"8";
      TAB 26;"10";TAB 28;"15"
940 PRINT TAB 27;"[KMC]"
950 PLOT 0,19: DRAW 235,0
960 PLOT 19,0: DRAW 0,170
970 FOR i=4 TO 23 STEP 2
980 PLOT 10*i-1,19: DRAW 0,3
990 NEXT i
1000 FOR j=4 TO 16 STEP 2
1010 PLOT 19,10*j-1: DRAW 3,0
1020 NEXT j
1030 REM
1040 REM Dane do skalowania osi x-y
1050 READ k1,k2,k3,k4,k5,k6,k7,k8,k9
1060 DATA 1,40,1,1,0,4,19,06,19
1070 FOR k=k1 TO k2 STEP k3
1080 LET g=k*k4+k5
1090 LET g1=g*k6+k7
1100 LET k00=k/2
1110 LET k0=INT(k00)
1120 LET k0=2*k0
1130 IF k0<k THEN GO TO 1160
1140 LET r2=(g)*k0+k9
1150 CIRCLE g1,r2,2
1160 LET dt=a*a*g
1170 LET d2=dt*k8+k9
1180 PLOT g1,d2
1190 NEXT k
1200 STOP
```

W załączonym obok programie na popularny komputer ZX Spectrum 48K firmy Sinclair ma to miejsce w pętli wewnętrznej rozpoczynającej się w linii 350, a kończącej w linii 610. Mianowicie w pojedynczym kroku, w linii 360, losowany jest kąt biegunowy f z przedziału $[0, 2\pi)$. Ten krok to właśnie pojedynczy krok Monte Carlo (kMC), który traktuje się tutaj jako umowną jednostkę czasu. Kolejne jednostki indeksowane są zmienną sterującą t przyjmującą wartość początkową $t = 1$ i końcową $t = tk$; tk wprowadzono z zewnątrz w linii 90 w postaci danej liczbowej.

Obsługa programu jest łatwa, a mianowicie: po bezbłędnym wpisaniu programu do mikrokomputera i uruchomieniu go klawiszem RUN należy po prostu stosować się do wyświetlanych na ekranie poleceń umieszczonych w nawiasach.

Uruchomienie programu na innym mikrokomputerze wymaga, niestety, modyfikacji. Na przykład mogą ulec zmianie instrukcje graficzne PLOT i DRAW lub funkcja RND, a także może ulec zmianie rola znaków interpunkcyjnych.

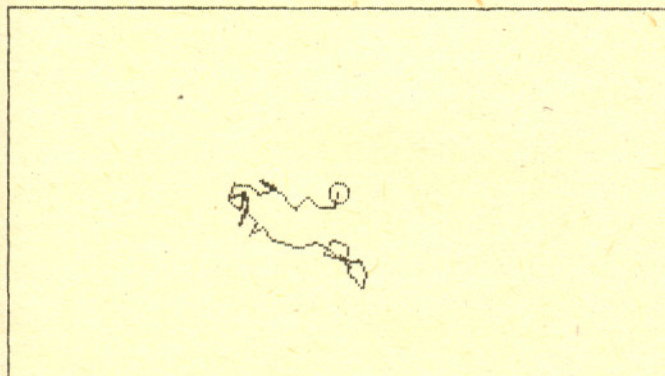
Następnie, w linii 370 obliczana jest x -owa składowa elementarnego przemieszczenia cząstki, oznaczona tutaj przez dx , oraz x -owa składowa nowego, aktualnego wektora położenia cząstki, oznaczona przez $x1$. Analogicznie obliczane są w linii 380 odpowiednie składowe y -owe (zauważmy, że wektor poprzedniego położenia cząstki oznaczono przez $(x0, y0)$). W ten sposób błądząca cząsteczka przemieszcza się z danego położenia $(x0, y0)$ w sąsiednie położenie $(x1, y1)$, przy czym prawdopodobieństwo tego przemieszczenia na jednostkę czasu (v), zwane także częstością przemieszczeń, przyjęto równe 1 kMC. (Fakt, że czas komputerowy t ma charakter dyskretny, nie ma w tym jednocząstkowym przypadku większego znaczenia.)

W wewnętrznej pętli konstruowane jest więc indywidualne „doświadczenie” komputerowe polegające na generowaniu dwuwymiarowego błędzenia przypadkowego pojedynczego punktu, który kreśli trajektorię swego ruchu na ekranie monitora.

Rysunki 1a i 1b stanowią kopię ekranu odpowiednio po pierwszym i czterdziestym kMC. Trajektoria tego ruchu jest, jak należało oczekiwać, linią łamaną złożoną z chaotycznie ukierunkowanych prostych odcinków. W każdym indywidualnym „doświadczeniu” komputer oblicza (linie 580 i 590) zależny od czasu kwadrat przesunięcia $[\Delta R(t)]^2 = (\sum_{j=1}^N \Delta r_j)^2$.

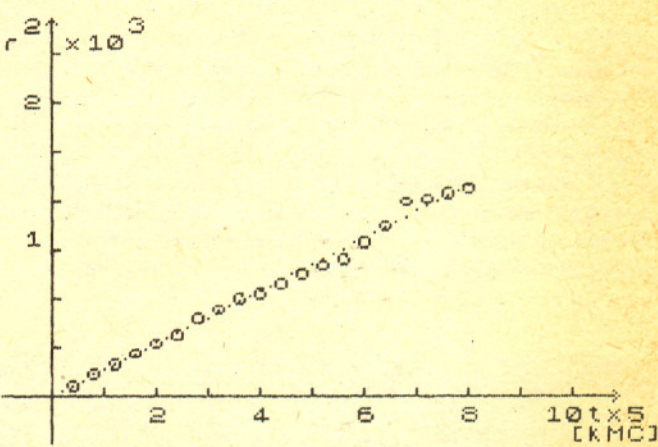
Następnie, między linią 620 a 640, uśrednia tę wielkość po utworzonym zespole statystycznym niezależnych „doświadczeń” (tzn. oblicza średnią arytmetyczną tej wielkości po liczbie „doświadczeń”), dając w efekcie zależne od czasu średnie przesunięcie kwadratowe $\langle [\Delta R(t)]^2 \rangle$. Wyniki otrzymane

```
nr exper.=2;liczba exper.=100
```



Rys. 1a. Kopia ekranu monitora po wykonaniu przez komputer drugiego „eksperymentu” należącego do zespołu statystycznego. Linia łamana pokazuje trajektorię cząstki Browna do chwili $t = 40$ kMC (kółko na środku rysunku oznacza miejsce startu w chwili $t = 0$).

z symulacji dla każdej chwili czasu t zestawiono w tabelce, którą komputer wyświetla na ekranie po zakończeniu „eksperymentu” (zależnie od czasu średnie przesunięcie kwadratowe oznaczono przez $\langle r^2(t) \rangle$). Ponadto wyniki te prezentuje w postaci wykresu (kółka na rys. 2) porównując je z przewidywaniami teoretycznymi (kropki na tym samym rysunku).



Rys. 2. Średnie przesunięcie kwadratowe $\langle [\Delta R(t)]^2 \rangle$ (oznaczone skróto przez r^2) jako funkcja czasu t . Kółkami zaznaczono wynik „doświadczenia” komputerowego, a kropkami dla porównania rezultat teoretyczny.

Można wykazać, że na mocy statystycznej niezależności elementarnych przemieszczeń zachodzi następująca zasadnicza relacja:

$$\langle [\Delta R(t)]^2 \rangle = \nu a^2 t,$$

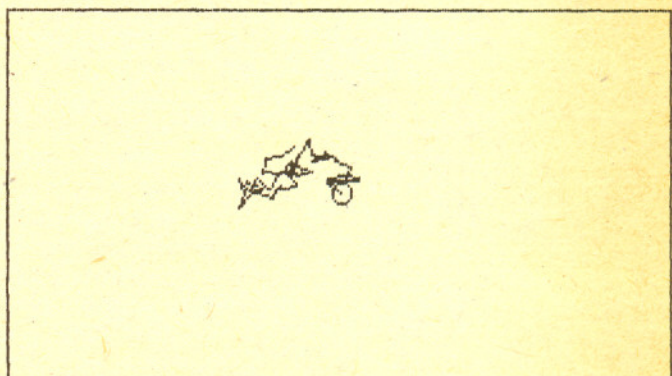
gdzie $\nu = N/t$. Wielkość tę można interpretować, dla dużych czasów t , jako średnią liczbę elementarnych przemieszczeń w jednostce czasu (czyli częstość przemieszczeń). W powyższym wzorze występuje charakterystyczna własność problemu: średnia z kwadratu N elementarnych przypadkowych niezależnych przemieszczeń jest proporcjonalna do N , a nie do N^2 . Stąd współczynnik dyfuzji D swobodnej cząsteczki Browna dany jest wzorem

$$D = \nu a^2 / 2d,$$

gdzie d jest wymiarem przestrzeni (w naszym przypadku, oczywiście, $d = 2$). Właśnie powyższe dwie relacje są sprawdzane w omawianym „doświadczeniu” statystycznym.

Wykres skonstruowano dla zespołu statystycznego składającego się z $s = 100$ indywidualnych „doświadczeń”, co, jak widać, zapewnia zadowalającą zgodność rezultatów „doświadczalnych” z przewidywaniami teoretycznymi. Liczba s jest daną wprowadzoną w linii 90 i w istocie rzeczy odpowiedzialną za rozrzut statystyczny punktów „doświadczalnych”. Przyjęcie np. $s = 400$ znakomicie zmniejszy rozrzut wyników, wydłuży jednak znacznie czas „eksperymentu” komputerowego. Wydaje się więc,

```
nr exper.=100;liczba exper.=100
(Datej dowolny klawisz)
```



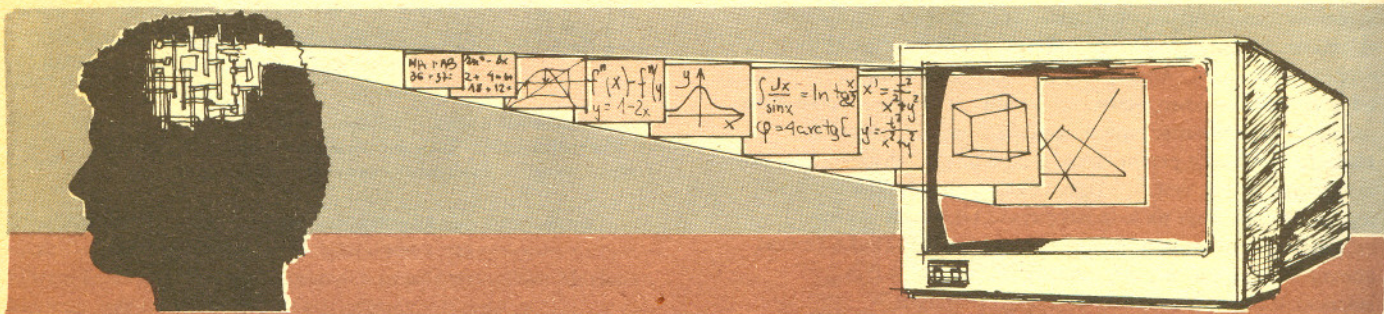
Rys. 1b. Analogiczna kopia ekranu po wykonaniu przez komputer ostatniego „eksperymentu”.

że $s \approx 100$ jest rozsądnym kompromisem — czas „eksperymentu” nie przekracza bowiem w takim przypadku 25 minut. Oczywiście komputer mógłby obliczać także i inne, niemniej ważne, charakterystyki ruchów Browna i porównywać je z teorią, ale tutaj chodziło jedynie o naszkicowanie idei, a nie o wyczerpującą analizę.

Jeżeli niektórzy Czytelnicy chcieliby we własnym zakresie dokonać modyfikacji zamieszczonego programu, to warto dodatkowo uwzględnić np. stochastyczny charakter długości elementarnego przemieszczenia. W tym celu należy wprowadzić linię przykładowo o numerze 365, w postaci: $LET I = -a * LN (RND)$ oraz w linii 370 i 380 dokonać zamiany zmiennej a na wyznaczoną powyżej długość elementarnego przemieszczenia l . Myślę, że Czytelnicy rozumieją, jakiemu (unormowanemu) rozkładowi prawdopodobieństwa podlega zmienna l . Należy zaznaczyć, że

przy takim podejściu wpływ warunków brzegowych (odbicie cząstki), przyjętych w programie, może być już istotny. Wymaga to więc ich zamiany na tzw. okresowe warunki brzegowe.

Innym, ambitnym zagadnieniem, które mógłbym Ci zaproponować, Szanowny Czytelniku, do rozwiązania, może być np. wyznaczenie prawdopodobieństwa powrotu cząsteczki po czasie t do punktu startowego. Prawdopodobieństwo to jest jedną z głównych charakterystyk błędzenia przypadkowego. Warto dodać ponadto, że o ile ruchy Browna pojedynczej cząsteczki są wystarczająco przebadane, o tyle zagadnienie ruchów Browna wielu oddziałujących wzajemnie ze sobą cząsteczek w skondensowanej zawieszynie jest obecnie przedmiotem intensywnych badań i wydaje się, że właśnie symulacje komputerowe (zwłaszcza metodą Monte Carlo) mogą przyczynić się do ich głębszego zrozumienia.



Zadania

Redaguje dr Rafał SZTENCEL

M 445. Czy szachownicę o wymiarach 10×10 można pokryć kostkami domina o wymiarach 2×1 w taki sposób, by dokładnie połowa z nich była ustawiona poziomo?

Rozwiązanie na str. 14

M 446. Wykazać, że czworokąty wypukłe o tych samych środkach boków mają równe pola.

Rozwiązanie na str. 2

M 447. Startująca z zera cząstka błądzi losowo po prostej w taki sposób, że z prawdopodobieństwem p wykonuje krok o jednostkę w prawo, z prawdopodobieństwem $q = 1 - p$ w lewo; kolejne kroki są niezależne. Niech S_n oznacza położenie cząstki po n krokach.

Obliczyć $E \left(\frac{q}{p} \right)^{S_n}$.

Rozwiązanie na str. 7

Redaguje mgr Rafał STAROŃSKI

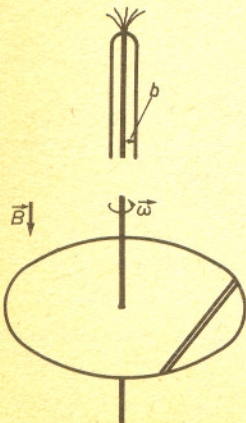
F 204. Piorunochron składa się ze sztywnego prostoliniowego przewodnika i otaczającej go metalowej rurki o promieniu $b = 1$ cm (rys. 1). W wyniku przepływu prądu wyładowania atmosferycznego rurka uległa zgnieceniu. Wyjaśnij dlaczego? Oszacuj wielkość ładunku elektrycznego, jaki przepłynął przez piorunochron, jeśli wiadomo, że rurka może wytrzymać ciśnienie nie większe niż 30 kPa, a czas przepływu prądu wynosi 10^{-3} s.

Rozwiązanie na str. 6

F 205. Płaska tarcza o promieniu R , wykonana z dielektryka, wiruje ze stałą prędkością kątową ω w stałym polu magnetycznym B skierowanym prostopadle do tarczy. Na powierzchni tarczy przymocowano wzdłuż cięciwy przewodnik o długości l (rys. 2). Obliczyć różnicę potencjałów między końcem przewodnika i jego środkiem.

Rozwiązanie na str. 3

Rys. 1



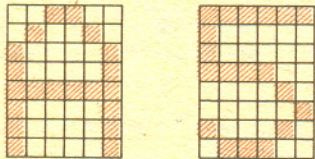
Rys. 2

Rysowanie prostej nie jest proste

Dr Michał JANKOWSKI

Narysowanie prostej przechodzącej przez dwa dane punkty na płaszczyźnie może sprawić kłopoty co najwyżej przedszkolakom. Czy to zadanie będzie równie łatwe, gdy kartkę papieru zamienimy na monitor (ekran) komputera?

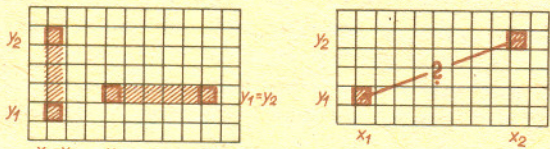
Wszystko, co oglądamy na monitorze, a więc litery, cyfry i linie, powstaje przez wyświetlenie wyglądających jak małe kropki najmniejszych części ekranu nazywanych z angielskiego pikselami.



Rys. 1

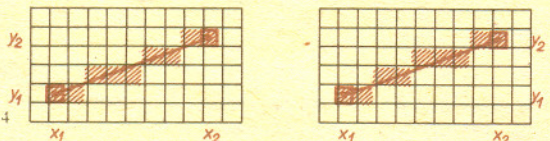
Na rysunku 1 pokazujemy w powiększeniu, jak z pikseli można zbudować literę A i cyfrę 5. Czytelnicy, którzy znają ZX Spectrum, wiedzą, że dla tego mikrokomputera ekran dzieli się na 176 wierszy i 256 kolumn, czyli składa się z 45056 pikseli.

Profesjonalne monitory używane w grafice komputerowej mają znacznie większą rozdzielczość, np. 4096 x 4096, i bardzo dużo kolorów, ale i odpowiednio wyższą cenę. Te urządzenia, tak jak i zwykle odbiorniki TV, są przykładami tzw. rastrowych urządzeń graficznych. Rysowanie na nich możemy porównać do rysowania na kartce z zeszytu w kratkę, gdzie zamiast robienia cienkich kresek wolno nam tylko zamalowywać całe kratki. Zadanie narysowania odcinka wygląda teraz zupełnie inaczej. Dwie ustalone kratki (albo piksele ekranu) o współrzędnych całkowitych równych numerom wiersza i kolumny mamy połączyć zamalowując odpowiednie kratki (albo piksele) między nimi. Rozwiązanie jest oczywiste, jeśli oba końce odcinka leżą w tej samej kolumnie lub w tym samym wierszu (rys. 2a i 2b). W innych



Rys. 2

przypadkach, na przykład takim jak na rysunku 3, końcowe kratki (piksele) odcinka możemy łączyć różnymi „schodkami” (rys. 4) i często trudno powiedzieć, które z połączeń jest lepsze.



Rys. 4

Spróbujmy podać ogólne rozwiązanie. Niech (x_1, y_1) i (x_2, y_2) będą końcami rysowanego odcinka. Przy klasycznym rysowaniu na płaszczyźnie byłyby to punkty o tych współrzędnych, przy komputerowym rysowaniu na ekranie numery wierszy i kolumn, w których leżą odpowiednie piksele, a wreszcie w naszym modelu z kartką papieru w kratkę są to współrzędne (tak, jak w grze w okręty) końcowych kratek odcinka. Zakładamy, że $x_1 < x_2$. Równanie odcinka możemy zatem zapisać w postaci

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1), \quad x \in [x_1, x_2]$$

lub

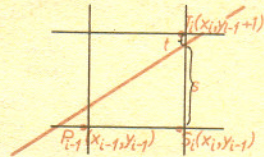
$$y = \frac{dy}{dx} x + b = ax + b, \quad x \in [x_1, x_2],$$

gdzie wprowadziliśmy oznaczenia

$$dy = y_2 - y_1, \quad dx = x_2 - x_1, \quad a = \frac{dy}{dx}, \quad b = y_1 - \frac{dy}{dx} x_1.$$

Narzuca się stąd następujący algorytm rysowania odcinka: — obliczamy wartości rzeczywiste a i b , dla kolejnych wartości x zmieniających się co 1 od x_1 do x_2 : — obliczamy liczbę rzeczywistą $r = ax + b$, — zaokrąglamy r do liczby całkowitej y , — zamalowujemy kratkę (wyświetlamy piksel) o współrzędnych (x, y) .

Istotną wadą tego algorytmu jest wykonywanie działań na liczbach rzeczywistych, co w większości komputerów jest znacznie kosztowniejsze (tzn. zajmuje dużo więcej czasu) niż działania na liczbach całkowitych. A przecież zarówno końce odcinka, jak i wszystkie zamalowywane kratki mają współrzędne całkowite. Własność tę wykorzystuje algorytm Bresenhama, który niżej opiszemy. Zakładamy dla uproszczenia, że współczynnik kierunkowy odcinka — a — jest zawarty między 0 i 1. Na rysunku 5 punkt P_{i-1} jest środkiem kolejnej kratki (piksla) tworzącej rysowany odcinek. W następnym kroku wybieramy tę kratkę,



Rys. 5

której środek leży w pionie bliżej rzeczywistego odcinka (narysowanego kolorem). Warunek ten możemy sformułować następująco: jeśli $s - t < 0$, to wybieramy kratkę o środku S_i , a w przeciwnym razie kratkę o środku T_i (inne kratki nie wchodzi w grę, bo $0 < a < 1$). Idea algorytmu jest bardzo prosta, ale czy można podejmować decyzję o wyborze S_i lub T_i wykonując tylko działania na liczbach całkowitych? Aby się o tym przekonać, zauważmy (patrz rys. 5), że

$$s = \frac{dy}{dx} (x_i - x_1) - (y_i - y_1), \quad t = 1 - s,$$

a stąd

$$dx \cdot (s - t) = 2dy \cdot x_i - 2dx \cdot y_{i-1} - 2dy \cdot x_1 + 2dx \cdot y_1 - dx.$$

Oznaczmy tę wielkość przez d_i . Ponieważ $dx > 0$, więc znak $d_i = dx \cdot (s - t)$ decyduje o wyborze następných kratek. Między d_{i+1} i d_i zachodzi zależność

$$d_{i+1} - d_i = 2dy \cdot (x_{i+1} - x_i) - 2dx \cdot (y_i - y_{i-1}),$$

ale $x_{i+1} - x_i = 1$, a więc

$$d_{i+1} = d_i + 2dy - 2dx \cdot (y_i - y_{i-1})$$

i, jak łatwo sprawdzić,

$$d_i = 2dy - dx.$$

Potrąfimy zatem obliczać kolejne d_i i możemy już zapisać algorytm Bresenhama. Ma on następującą postać:

— obliczamy $d_1 = 2dy - dx$,

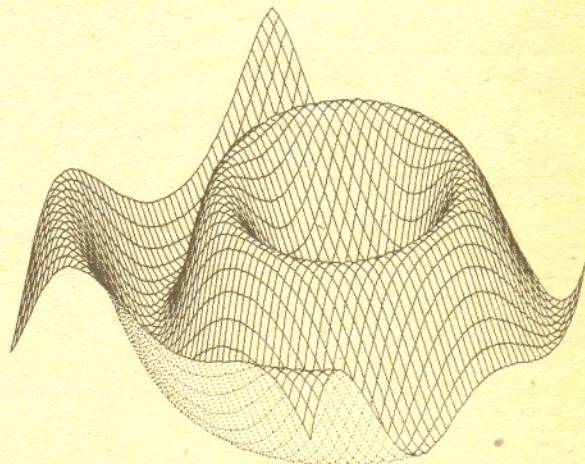
dla $i = 1, 2, \dots, x_2 - x_1$:

— jeśli $d_i < 0$, to {wybieramy jako następną kratkę o środku S_i ,

wtedy $y_i = y_{i-1}$ i obliczamy $d_{i+1} = d_i + 2dy$,

— w przeciwnym razie {wybieramy jako następną kratkę o środku T_i , wtedy $y_i = y_{i-1} + 1$ i obliczamy $d_{i+1} = d_i + 2(dy - dx)$.

Jeśli przebrnąłeś Czytelniku przez te proste (?) rachunki dla rysowania prostych, to spróbuj się zastanowić nad rysowaniem (oczywiście na urządzeniu rastrowym) na przykład okręgu lub wykresów takich powierzchni jak przedstawiona na rysunku 6.




Rys. 6

Małe bazy danych w mikrokomputerze

Mgr Jan RASZEWSKI

Baza danych — plik lub pliki danych
(ewentualnie ze zbiorami indeksów)
zawierające całość przetwarzanych informacji.

Plik danych — zbiór rekordów
zawierających opis obiektów tego samego
rodzaju.



Rozwiązanie zadania M 445. Nie można ponumerować pól szachownicy liczbami od 1 do 100 w taki sposób, by w k -tym wierszu znalazły się kolejno liczby $10(k-1)+1$, $10(k-1)+2$, ..., $10k$. Dwadzieścia pięć kostek domina ustawionych poziomo pokrywa wtedy dwadzieścia pięć liczb parzystych i dwadzieścia pięć nieparzystych. Kostka pionowa pokrywa parzystą ilość liczb nieparzystych (0 lub 2). Kostki pionowe nie mogą więc pokryć pozostałych dwudziestu pięciu liczb nieparzystych.

Jednym z głównych zastosowań informatyki, od początku jej istnienia, było gromadzenie i przetwarzanie wielkich ilości informacji. Obecnie trzeba tworzyć bazy danych tak duże, że przeszukiwanie ich przekracza możliwości najpotężniejszych komputerów. Jedyne wyjście polega na tym, żeby pamiętać tylko wybrane, podstawowe informacje. Pozostałe komputer powinien umieć wydedukować (pisał o tym J. Pacholczyk w *Delcie* 1/1986). Tworzenie małych baz danych nigdy nie było zadaniem informatyki. Aż do czasu, kiedy pojawiły się mikrokomputery.

Czy mikrokomputer w ogóle nadaje się do tego celu? To zależy od jego parametrów. A zwłaszcza od rozmiaru pamięci wewnętrznej oraz pojemności pamięci masowej. Im większa pamięć wewnętrzna, tym bardziej zaawansowane może być oprogramowanie. Im większa pamięć masowa, tym bardziej rozbudowana może być sama baza.

Do obsługi niewielkich baz danych najczęściej wykorzystywany jest system dBASE, rozpowszechniony w dwóch wersjach. Wersja II powstała w 1982 roku dla maszyn zawierających co najmniej 64 kB pamięci operacyjnej i jeden napęd dysków elastycznych. W 1984 roku z myślą o komputerach szesnastobitowych powstała wersja III, która wymaga 256 kB oraz dwóch dysków. dBASE III jest jednak tak rozbudowany, że staje się wygodny dopiero przy współpracy z dyskiem stałym typu Winchester. Wymagania te spełnia coraz bardziej popularny, również w Polsce, profesjonalny mikrokomputer IBM PC/XT.

Najistotniejszym problemem jest szybka odpowiedź programu na zadane pytanie, czyli odpowiedni algorytm wyszukiwania informacji w bazie danych. Wyobraźmy sobie, że mamy bazę danych zawierającą dane bibliograficzne na temat artykułów i że chcemy znaleźć artykuł prof. Trąbalskiego. Jak to zrobić najłatwiej? Ano, po kolei: może pierwszy artykuł? Jeśli nie, to może drugi itd. Średnio przeszukać trzeba około połowy bazy danych. A teraz wyobraźmy sobie, jak byśmy szukali, gdyby baza danych była uporządkowana alfabetycznie. Przede wszystkim nie zaczęlibyśmy od początku, a od środka. Po pierwszym porównaniu wiedzielibyśmy, czy szukany autor znajduje się w pierwszej, czy w drugiej połowie. Po drugim porównaniu mielibyśmy oszacowane jego położenie z dokładnością do ćwiartki itd. Łatwo widać, że przy takim postępowaniu znajdziemy odpowiedź NAJPOŹNIEJ po $\log_2 n$ porównań, gdzie n to liczba artykułów. Nie jest to duża liczba. Dla tysiąca wynosi 10, dla miliona — 20, dla miliarda (a, jak sądzę, nie napisano jeszcze tylu artykułów) — 30. Metoda ta nazywa się *przeszukiwaniem binarnym*.

A więc mamy cudowną metodę na błyskawiczne wyszukiwanie? Niezupełnie. Pamiętajmy, że zrobiliśmy jedno niebagatelne założenie — że dane są uporządkowane. Niestety, algorytmy sortowania są bardzo złożone (patrz *Delta* 9/1985, artykuł T. Przytyckiej). W praktyce opłaca się sortować tylko takie bazy danych, w których zmiany są wprowadzane bardzo rzadko. Poza tym zwróćmy uwagę, że samo sortowanie nie zawsze rozwiązuje problem. Skoro mamy dane posortowane według nazwisk autorów, to nie są one posortowane według roczników. Nie da się więc już wykorzystać naszego szybkiego algorytmu wyszukiwania do znalezienia artykułów z określonego roku.

Są jednak inne metody. Najpowszechniej stosowana to korzystanie z tzw. *plików indeksowanych*. Proces indeksowania polega na tym, że tworzymy dodatkowy plik, zawierający informacje na temat położenia rekordów w pliku danych. Plik zawierający dane nie ulega zmianie. Natomiast para plików: dane + indeksy, jest widziana przez algorytm wyszukiwania tak, jakby dane były posortowane. Dla jednej bazy danych możemy założyć wiele różnych plików z indeksami. W ten sposób możemy udawać, że nasze dane są posortowane zarówno według nazwisk autorów, jak według roczników albo tytułów czasopism.

Zapyta ktoś: po co wyszukiwanie liniowe, skoro można tak szybko wyszukiwać po indeksach? Wyszukiwanie za pomocą indeksów jest jednak mniej uniwersalne. Można je stosować tylko wtedy, gdy odnosi się do pola danych, po którym wykonano indeksowanie. Jeśli chcemy przeszukiwać dane według bardziej złożonych kryteriów, musimy korzystać ze znacznie wolniejszego, ale ogólnego wyszukiwania liniowego. Dlatego właśnie dBASE oferuje dwie instrukcje wyszukiwania: wolno działającą LOCATE, której parametrem może być wyrażenie opisujące dowolne informacje występujące wśród danych lub związki między nimi, oraz bardzo szybko FIND, której parametrem może być tylko konkretna wartość pola danych, według którego plik był wcześniej indeksowany.

Jeśli na przykład rekord danych zawiera pola *Nazwisko* i *Rok wydania*, to instrukcje

```
FIND Trąbalski  
oraz  
LOCATE FOR Nazwisko = „Trąbalski”
```

obie znajdują kolejny artykuł Trąbalskiego, choć pierwsza jest wielokrotnie szybsza. Ale jeśli chcemy znaleźć kolejny artykuł Trąbalskiego z lat 1970—1976, to musimy się posłużyć instrukcją LOCATE FOR Nazwisko = „Trąbalski”. AND. Rok wydania > 1969 .AND. Rok wydania < 1977.

Projektując algorytmy wyszukiwania musimy pamiętać, że trzeba pogodzić sprzeczne cele. Podczas wprowadzania i modyfikowania danych trzeba aktualizować pliki z indeksami. A zatem im więcej plików indeksowanych, tym dłużej będzie trwało aktualizowanie informacji.

Oprogramowanie bazy danych artykułów, w którym brałem niedawno udział, jest typowym zagadnieniem z dziedziny małych baz danych. Zadanie w pierwszej chwili może się wydać banalne: stworzyć katalog artykułów naukowych dla pracowników jednej z pracowni na Wydziale Chemii UW. Jest ich pięciu, każdy ma 1000 do 2000 artykułów. Niektóre jednak się powtarzają, więc w sumie liczba różnych artykułów nie przekracza 6000. Dla każdego artykułu trzeba pamiętać: nazwiska autorów, tytuł artykułu, tytuł pisma, rok, tom, strony, kilka linijek streszczenia, dla łatwiejszej identyfikacji, kilka słów kluczowych oraz identyfikatory właścicieli artykułów. W sumie dla każdego artykułu niespełna pięćset znaków — a więc około 500 bajtów. Jeśli obsługiwać mamy 6000 artykułów, to musimy przyjąć na „zapas, że w ciągu kilku lat może być ich nawet 10000. Po przemnożeniu przez długość opisu artykułu okaże się, że baza danych zajmie 6 MB. Chemikom trochę zrzęda mina, gdy dowiedzieli się, że nie chodzi o kilka dyskietek, a o znaczną część 10 megabajtowego dysku stałego w ich mikrokomputerze IBM PC/XT.

Pole rekordu — opis pojedynczej informacji dotyczącej jednego obiektu, np. nazwisko autora, tytuł artykułu lub tytuł czasopisma.

Rekord — całkowity opis obiektu, w omawianym przykładzie opis jednego artykułu.

System baz danych (ang. DBMS — Data Base Management System) — baza danych wraz z programami, które zapewniają aktualizowanie oraz wszechstronne wyszukiwanie informacji.

nr	autorzy	tytuł artykułu	tytuł pisma	rok
1	ROSSI T.M. SHELLS D.	OPTIMIZATION OF A FLOW-INJECTION	Anal.Chem.	1982
2	LYNCH T.P. TAYLOR A.	FULLY AUTOMATIC FLOW-INJECTION	Analyst	1983
3	FOSSEY L. CHATWELL F.	CHARACTERIZATION OF SOLVENT EXTRACTION	Anal.Chem.	1982
4	HARRIS Z.M.	FLOW INJECTION OF ULTRATRACE	Anal.Chem.	1982
5	SECKSCHLAGER K.	STEPWISE INFORMATION THEORY APPROACH TO	Mikrochim.Acta	1981
6	BUSSEFELD B.	SUPPLYNOWY TYP ZAMRU DOZIACZEGO DO	Chem.Anal.	1981
7	BRIDGEMAN P.J.	THE BIO-ANALYTICAL POTENTIAL OF	Chem.Anal.	1982
8	HULANICKI A.	LEWENST DIFFUSION-LAYER MODEL FOR COPPER	Talanta	1976
9	FRIDOLEY D.	COMPARISON OF COPPER(II) ION-SENSITIVE	Anal.Chem.	1976
10	MINDY S.J. HASSORY W.	INTERFERENCE FILMS ON THE SENSITIVE	Analyst	1979
11	BERTRAM R. HILLIER	STIMULATION OF THE DISSOLUTION OF	Electroanal.Chem.	1981
12	FOGG A.S. CHASEI A.	FLOW INJECTION VOLTAMMETRIC DETECTION	Analyst	1983
13	HANSEN E.H. GHOSH A.	FLOW INJECTION ANALYSIS OF ENVIRONMENTAL	Analyst	1977
14	FRIDOLEY D.	SULFIDE AND ACID INTERFERENCES IN	Anal.Chem.	1982
15	SCAMPORLO A.	POTENTIOMETRIC MICRODETERMINATION OF	Mikrochim.Acta	1983

1 - 0-pocz 2 - o-nast 3 - p-popr 4 - od-rod 5 - f-doklad 6 - r-ile ar 7 - b-help 8 - a-pa 9 -

Rys. 1. Postać ekranu po wykonaniu dyrektywy „b”, wyświetlającej skrótowy opis pierwszych piętnastu artykułów.

nr	autorzy	tytuł artykułu	tytuł pisma	rok
8	HULANICKI A.	LEWENST DIFFUSION-LAYER MODEL FOR COPPER	Talanta	1976
12	HULANICKI A.	TROJANDZIE CHLOROWE COPPER MEMBRANE	Talanta	1976
15	HULANICKI A.	OKRĄŻY DETERMINATION OF COPPER IN CONCENTRATED	Anal.Chem.	1979
12	HULANICKI A.	TROJANDZIE DETERMINATION OF COPPER IN WATER RES.	Water Res.	1977
12	HULANICKI A.	KRĄŻY EFFECT OF SOME CHELATING LIGANDS	Anal.Chem.	1984

Podaj nazwisko szukanego autora: Hulaniccki A.

1 - k-klucz 2 - a-autor 3 - t-tytuł 4 - e-tytuł 5 - p-pocz 6 - o-nast 7 - p-popr 8 - f-doklad 9 - a

Rys. 2. Za pomocą dyrektywy „a” zostały wyświetlone skrótowe informacje o wszystkich artykułach napisanych przez prof. A. Hulanicckiego.

Program pozwala korzystać z następujących dyrektyw:

- menu -
- 1 - wprowadzanie artykułów do bazy danych,
 - d - usuwanie artykułów o danej numerze,
 - a - wprowadzanie poprawek do artykułu o danej numerze,
 - f - wpisanie pełnych informacji o artykule
 - s - alfabetyczne spisy kluczy i/lub tytułów czasopism,
 - l - odzyskanie pamięci po usuniętych artykułach,
 - h - to jest właśnie to, co teraz czytasz,
 - e - chwilowe rozłączenie z obrotowym komputerem,
 - l - wyswietlenie pierwszego menu,
 - 2 - wyswietlenie drugiego menu.
- !! Jeśli chcesz poznać dyrektywy proponowane w innym menu nacisnij odpowiedni klawisz 1, 2 lub 3.
!! Wskazanie dowolnego innego klawisza spowoduje powrót do programu.

Rys. 3. W każdym menu można uzyskać dokładne informacje o tym, w jaki sposób korzystać z systemu. Wystarczy nacisnąć literę h — jak HELP, czyli: pomocy!

Pierwszą rzeczą, jaką należy zrobić przystępując do pracy nad bazą danych, jest określenie jej struktury. Trzeba więc ustalić, co będzie opisywał pojedynczy rekord (w tym przypadku — jeden artykuł), oraz jakie będą pola rekordu. Następna czynność to ustalenie operacji, jakie będą wykonywane. Każdy system baz danych musi zapewniać realizację podstawowych operacji, takich jak wprowadzanie, modyfikowanie, usuwanie i, oczywiście, wyszukiwanie danych. Jak już widzieliśmy, najistotniejszy jest problem wyszukiwania informacji i pod jego kątem należy projektować pozostałe elementy systemu. W naszym systemie szybkie wyszukiwanie odbywa się według nazwisk autorów oraz słów kluczowych. Nieco wolniejsze — według koniunkcji autorów i słów kluczowych. Natomiast z metody liniowej korzystają najbardziej złożone sposoby wyszukiwania, np. według konkretnego słowa występującego w tytule artykułu.

Osobnym zagadnieniem jest zorganizowanie sprawnej i wygodnej komunikacji z użytkownikiem systemu. Już w tak prostym, jak omawiany tutaj, systemie użytkownik dostał do dyspozycji około dwudziestu dyrektyw. Nie należy wymagać, żeby uczył się ich nazw albo skrótów na pamięć. Najkorzystniejszą metodą jest zaprojektowanie tzw. systemu sterowanego przez menu. Użytkownik powinien mieć zawsze na ekranie wyświetlone podstawowe możliwości systemu, wraz ze sposobem ich realizacji. W systemie, o którym mówimy, użytkownik ma do dyspozycji trzy różne menu. W momencie uruchomienia systemu zgłasza się pierwsze, które pozwala na wyświetlanie skrótowych informacji o piętnastu wybranych artykułach naraz lub pełnych informacji o wybranym artykule (rys. 1). Drugie menu służy do wyszukiwania informacji i wyświetlania na różne sposoby informacji znalezionych (rys. 2). Trzecie menu służy do modyfikowania zawartości bazy danych — a więc wprowadzania, poprawiania i usuwania opisów artykułów. Z każdego menu dostępna jest dyrektywa HELP, która informuje o działaniu poszczególnych dyrektyw (rys. 3).

Jak widać, nawet tak pozornie proste zadanie wymaga dość złożonego oprogramowania. Napisanie go zajęło dwóm osobom kilka tygodni, a trwałoby to bez porównania dłużej, gdybyśmy nie korzystali z pomocy mechanizmów dostarczanych przez dBASE. System składa się z 37 programów zajmujących łącznie około 40 kB tekstu.

dBASE III to jednak coś więcej niż tylko mechanizmy tworzenia i przeszukiwania baz danych. Jest to tzw. pakiet zintegrowany. Idea takiego pakietu polega na dostarczeniu w jednym programie kompletu narzędzi potrzebnych do pisania i uruchamiania programów. Mamy więc bardzo wygodny (tzw. *full-screen*, operujący na całym ekranie) procesor tekstów, pozwalający na pisanie i modyfikowanie programów, które mogą być natychmiast wykonane przez interpreter. A jednocześnie mamy do dyspozycji dyrektywę systemu operacyjnego, które pozwalają np. na kopiowanie plików albo ich usuwanie. Język programowania dBASE III jest łatwy do opanowania i dość wygodny w użyciu, choć może razić jego „gadatliwość”. Na szczególną uwagę zasługują prostota operacji wykonywanych na ekranie — a zwłaszcza czytania i pisania w dowolnym punkcie. Język programowania zawiera ponadto niesłychanie silny mechanizm makrodefinicji, dzięki któremu można dynamicznie tworzyć zmienne, przechowywać nazwy zmiennych na innych zmiennych albo dynamicznie, w trakcie działania programu, tworzyć tekst wyrażenia logicznego, według którego następnie będziemy wyszukiwać informacje.

Warto dodać, że system dBASE III należy do najdrożej sprzedawanych pakietów. Jest też zabezpieczony przed „piractwem”: próba pracy z oprogramowaniem skopiowanym kwitowana jest komunikatem „nieautoryzowana kopia” i uszkodzeniem kopii oraz okolicznych danych.

Czołówka ligi zadaniowej "Klub 44 M"
po uwzględnieniu ocen rozwiązań
zadań 127 /WT=2,25/ i 128 /WT=3,16/
z numeru 3/1986

Klub 44

Kazimierz Serbin	- Sanok	44,79pkt
Marian Roman	- Eżk	43,09pkt
Andrzej Sudoł	- Nowy Sącz	42,56pkt
Marcin Mazur	- Białystok	42,06pkt
Tomasz Rawlik	- Gliwice	41,48pkt
Zbigniew Koza	- Jelenia G.	41,32pkt
Marek Prauza	- Porań	41,30pkt

Pan Serbin po raz drugi przekracza linię 44 punktów.

Czołówka ligi zadaniowej "Klub 44 P"
po uwzględnieniu ocen rozwiązań
zadań 25 /WT=2,61/ i 26 /WT=2,27/

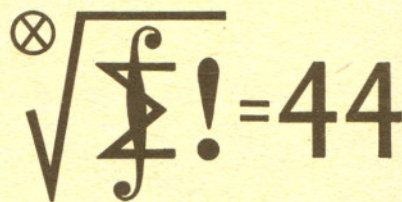
Tomasz Rawlik	- Gliwice	33,80pkt
Aleksander Surma	- Myszków	25,72pkt
Dzierżyszław Lipniacki-Lublin		23,47pkt

Liga zadaniowa Wydziału Matematyki, Informatyki i Mechaniki, Wydziału Fizyki Uniwersytetu Warszawskiego i Redakcji *Delty*

Skrót regulaminu

Każdy może nadsyłać rozwiązania zadań z numeru n w terminie do końca miesiąca $n+2$. Szkice rozwiązań zamieszczamy w numerze $n+4$. Można nadsyłać rozwiązania czterech, trzech, dwóch lub jednego zadania (każde na oddzielnej kartce), można to robić co miesiąc lub z dowolnymi przerwami. Rozwiązania zadań z matematyki i z fizyki należy przysyłać w oddzielnych kopertach, umieszczając na kopercie dopisek: **Klub 44 M** lub **Klub 44 F**. Oceniamy zadania w skali od 0 do 1 z dokładnością do 0,1. Ocenę mnożymy przez współczynnik trudności danego zadania: $WT = 4 - 3S/N$, gdzie S oznacza sumę ocen za rozwiązania tego zadania, a N — liczbę osób, które nadesłały rozwiązanie choćby jednego zadania z danego numeru w danej konkurencji (M lub F) — i tyle punktów otrzymuje nadsyłający. Po zgromadzeniu 44 punktów, w dowolnym czasie i w którejkolwiek z dwóch konkurencji (M lub F), zostaje on członkiem Klubu 44, a nadwyżka punktów jest zaliczana do ponownego udziału. Trzykrotne członkostwo — to tytuł **Weterana**. Szczegółowy regulamin został wydrukowany w numerze 1/1986.

Termin nadsyłania rozwiązań: 30 XI 1986



Zadania z matematyki nr 135, 136

Redaguje dr Marcin E. KUCZMA

135. Okręgi K i L są wpisane w ramiona kąta o wierzchołku O . Dany jest ponadto trzeci okrąg styczny zewnętrznie do okręgów K i L odpowiednio w punktach P i Q . Dowieść, że punkty O, P, Q są współliniowe.

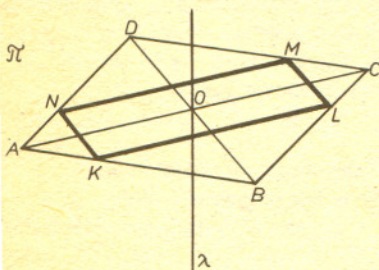
136. Udowodnić, że iloczyn dowolnych czterech
a) kolejnych liczb naturalnych,
b) kolejnych liczb parzystych,
c) kolejnych liczb nieparzystych
można przedstawić na dwa różne sposoby jako różnicę kwadratów dwóch liczb naturalnych.
Zadanie 136 przysłał pan Jerzy Janowicz z Bolesławca

Rozwiązania zadań z matematyki z numeru 5/1986

Przypominamy treść zadań:

131. Przez środki dwóch skośnych krawędzi czworokątnu poprowadzono płaszczyznę, rozcinając ją czworokątnem na dwie części. Jaki może być stosunek objętości tych części?

132. Udowodnić nierówność $(\sum_i \prod_j a_{ij})^k \leq \prod_j \sum_i a_{ij}^k$ dla $a_{ij} > 0$ ($i = 1, \dots, n$; $j = 1, \dots, k$).



131. Stosunek ten równa się 1, obie części mają równą objętość. Aby to wykazać, weźmy dowolną płaszczyznę π równoległą do obu wyróżnionych krawędzi skośnych i przecinającą czworokątnem (będzie to płaszczyzna rysunku) i rzutujemy na nią czworokątnem równoległe do prostej łączącej środki tych krawędzi. Obrazem czworokątnu będzie równoległobok $ABCD$, obrazem wspomnianej prostej — punkt O . Płaszczyzna σ , którą przecięto czworokątnem, przechodzi przez środki rozważanych dwóch krawędzi, zatem jej obrazem (w tym rzucie równoległym) jest prosta λ przechodząca przez O . Weźmy teraz pod uwagę przekrój czworokątnu płaszczyzną π . Krawędzie przecięcia płaszczyzny π ze ścianami czworokątnu są równoległe do odcinków AC i BD , zatem przekrój czworokątnu płaszczyzną π jest równoległobokiem ($KLMN$ na rysunku), a O jest jego środkiem symetrii. Prosta λ dzieli ten równoległobok na dwie części o równych polach. Płaszczyzna π była dowolnie wybrana z rodziny wszystkich płaszczyzn równoległych do wyróżnionych krawędzi i przecinających czworokątnem. Znaczy to, że każdy przekrój czworokątnu płaszczyzną z tej rodziny jest przez płaszczyznę σ rozcięty na części o równych polach. Na mocy zasady Cavalieriego płaszczyzna σ dzieli czworokątnem na części o równych objętościach.

132. Oznaczmy: $A_j = (\sum_i a_{ij}^k)^{1/k}$, $x_{ij} = A_j^{-1} a_{ij}$. Dla ustalonego i mamy

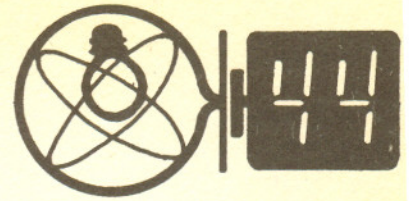
$$\prod_j a_{ij} = \prod_j A_j x_{ij} = \left(\prod_j A_j\right) \left(\prod_j x_{ij}\right) \leq \left(\prod_j A_j\right) \left(\frac{1}{k} \sum_j x_{ij}^k\right)$$

(średnia geometryczna i arytmetyczna). Sumując po $i = 1, \dots, k$ dostajemy

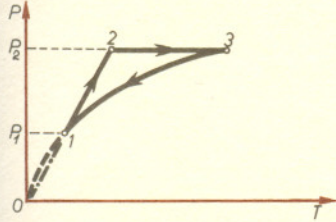
$$\sum_i \prod_j a_{ij} \leq \frac{1}{k} \left(\prod_j A_j\right) \left(\sum_{i,j} x_{ij}^k\right).$$

Ale $\sum_i x_{ij}^k = A_j^{-k} \sum_i a_{ij}^k = 1$, skąd $\sum_{i,j} x_{ij}^k = k$ i wobec tego $\sum_i \prod_j a_{ij} \leq \prod_j A_j$. Podnosząc stronami do k -tej potęgi otrzymujemy tezę zadania.

Redaguje dr Andrzej NADOLNY



33. Jednoatomowy gaz doskonały podlega odwracalnemu procesowi kołowemu, dla którego zależność ciśnienia p od temperatury T jest przedstawiona na wykresie (odcinek 1—3 opisywany jest zależnością $p = \text{const} \cdot \sqrt{T}$). Podać wykres zależności ciśnienia od objętości dla tego procesu i obliczyć sprawność silnika cieplnego realizującego ten cykl przy założeniu $p_2 = 2p_1$. Czy wynik dla gazu dwuatomowego byłby taki sam?



34. Posługując się argumentacją fizyczną ocenić, jaka część wokółsłonecznego toru Księżyca ma wypukłość zwróconą ku Słońcu. Niezbędne dane, w możliwie jak najmniejszej liczbie, należy wziąć z tablic.

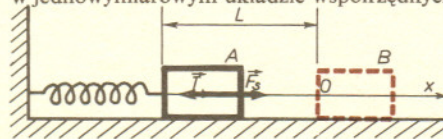
Rozwiązania zadań z fizyki z numeru 5/1986

Przypominamy treść zadań:

29. Klocek o masie m , spoczywający na płaskim, poziomym podłożu, jest połączony sprężyną o stałej sprężystości k ze stałym punktem, jak na rysunku. Sprężynę ściśnięto o odcinek L w stosunku do położenia swobodnego, a następnie puszczono. O jaki odcinek przesunie się klocek po podłożu (do swego pierwszego zatrzymania), jeżeli współczynnik tarcia klocka o podłoże wynosi f . Masę sprężyny należy zaniedbać.

30. Jaki co najmniej powinien być wypadkowy ładunek elektryczny Ziemi wraz z atmosferą, aby (przy założeniu kulisto-symetrycznego rozkładu tego ładunku) występowało elektrostatyczne „wymiatanie” z ziemskiego pola grawitacyjnego (z górnych warstw atmosfery) jednokrotnie dodatnio zjonizowanych atomów wszystkich pierwiastków. Jaką objętość powietrza w warunkach normalnych należałoby całkowicie zjonizować, aby łączny ładunek uzyskanych w ten sposób jonów N^+ i O^+ odpowiadał powyższemu ładunkowi? Niezbędne do obliczeń dane należy wziąć z tablic.

29. Ruch klocka odbywa się w kierunku działania sprężyny. Rozpatrujemy więc ten ruch w jednowymiarowym układzie współrzędnych przedstawionym na rysunku. Oznaczmy wychylenie



A — położenie „startowe” klocka
B — położenie klocka przy luźnej sprężynie

klocka z położenia „neutralnego” (swobodnej sprężyny) przez x , siłę działania sprężyny na klocek przez F_s , siłę tarcia przez T ; dodatnie wartości F_s oraz T odpowiadają wektorom tych sił zwróconym zgodnie z osią x . Z definicji stałej sprężystości k wynika $F_s = -kx$. W chwili startowej $x = -L$, co daje $F_s = kL$. Jeśli $kL \leq mg$, zachodzi $T = -F_s$ i klocek w ogóle nie ruszy z miejsca. W przypadku, gdy $kL > mg$, zacznie się ruch klocka. Wypadkowa siła działająca na klocek podczas tego ruchu będzie $F = F_s + T = -kx - fmg$. Ponieważ T do momentu pierwszego zatrzymania się klocka jest stałe, można siłę F przedstawić w postaci

$$F = -k(x - x_0), \text{ gdzie } x_0 = -\frac{fmg}{k}.$$

Ruch klocka na tym odcinku ma więc charakter ruchu harmonicznego względem położenia x_0 . Początkowe wychylenie klocka względem tego położenia wynosi $\Delta = -L - x_0$. Klocek zatrzyma się w położeniu $-\Delta$. Przesunie się on więc o odcinek

$$2|\Delta| = 2\left(L - \frac{fmg}{k}\right).$$

Taki sam wynik otrzymamy porównując pracę wykonaną na pokonanie siły tarcia do zmiany energii potencjalnej sprężyny (energia sprężyny rozciągniętej o l jest równa $kl^2/2$).

30. Elektrostatyczne „wymiatanie” naładowanych cząstek występuje wtedy, gdy siła odpychania elektrostatycznego $F_e = \frac{1}{4\pi\epsilon_0} \frac{Qq}{r^2}$ przewyższa siłę przyciągania grawitacyjnego $F_g = \gamma \frac{Mm}{r^2}$.

(ϵ_0 — przenikalność elektryczna próżni, Q — ładunek elektryczny Ziemi, q — ładunek cząstki, r — odległość od środka Ziemi, γ — stała grawitacyjna, M — masa Ziemi, m — masa cząstki).

Z warunku $F_e > F_g$ otrzymujemy $Q > \frac{4\pi\epsilon_0\gamma Mm}{q}$. Widać stąd, że przy jednakowym ładunku q

najtrudniej będą wymiatane ciężkie cząstki. Jako m podstawiamy więc masę atomu uranu ^{238}U , równą $4 \cdot 10^{-25}$ kg, jako q — ładunek elementarny (pozostałe wartości znane) i na poszukiwaną wartość krytyczną ładunku Ziemi uzyskujemy 0,1 C. Ładunek ten jest równy $7 \cdot 10^{17}$ ładunków elementarnych i taka powinna być liczba jonów N^+ i O^+ . Wymagana liczba atomów zawarta jest w $5 \cdot 10^{-10}$ kmola N_2 , O_2 — odpowiada to objętości gazu w warunkach normalnych równej $1 \cdot 10^{-8}$ m³. Wystarczyłoby więc całkowicie zjonizować zaledwie 10 mm³ powietrza, aby uzyskać omawiany efekt. Stąd widać, jak wielka jest koncentracja ładunków elektrycznych w przyrodzie i jak dokładnie są one skompensowane: wystarczy bardzo niewielkie odstępstwo od zrównoważenia ładunków przeciwnych znaków, aby wystąpiły kolosalne oddziaływania elektryczne (które tę równowagę przywracają).

